Instalación Cluster Openshift - UPI

13/03/2024

Version: 1.2

Fecha: 13-03-2024

Actores y roles en el requerimiento

Realizador	Empresa	Contacto
Jaime Galdames	INTAC	jgaldames@intaclatam.com
José Pablo Arancibia	INTAC	jparancibia@intaclatam.com

Historial de Cambios

Día de revisión	Revisado por	Razón de cambio
29/02/2024	INTAC	Primera versión del documento
11/03/2024	INTAC	Se agregan más especificaciones para los SO RHEL o CentOS, se añade configuración DHCP y DNS en Linux, se agregan comandos para generar Manifest Files, se agregan comandos para crear los custom initramfs para los worker y masters.
13/03/2024	INTAC	Se realizan cambios en configuraciones de DNS, DHCP, HAProxy. Se añaden más validaciones para los servicios TFTP y HTTP.

Historial de cambios documentos relacionados

Documentos relacionados

Observaciones

- Las configuraciones en este documento se realizaron en base a la instalación UPI Platform Agnostic (Cualquier plataforma). Ésta no difiere mucho con la instalación en VMWare, igualmente se añadieron las configuraciones para VMWare en paralelo.
- Se debe agregar más detalle a las configuraciones faltantes para la instalación UPI en VMWare.
- Todas las configuraciones de las máquinas mencionadas en este documento fueron probadas y ejecutadas con éxito, con excepción del ***"HAProxy Load Balancer"***.
- En el apartado de ***"Nodos"***, solo se instaló y configuró el nodo Bootstrap. Los master y workers deberían seguir los mismos pasos, pero con sus archivos correspondientes. De igual manera queda pendiente documentar el proceso para estas máquinas.
- Los sistemas operativos utilizados para la realizacion de este documento son: ***Windows Server 2016*** - ***Ubuntu v22.04*** - ***Centos 7***.
- Los CLI ***coreos-installer***, ***openshift-install*** y ***oc*** no son compatibles con versiones de Centos 7 hacia atrás. Por lo que es recomendable, si es que se usara RHEL o Centos, que se utilizen las versiones 8 y posteriores.

Pre-Requisitos.

- [x] Definir método de instalación. En este caso se utilizará el método UPI (User Provisioned Infrastructure).
- [x] Modelar infraestructura del cluster.
- [x] Descargar Sistema Operativo RHCOS según arquitectura y método de instalación.
- [x] Definir plataforma en donde se instalará el cluster.
- [ ] Habilitar puerto 443 en vCenter y los hosts ESXi.
- [ ] Permitir sitios requeridos para la instalación de OCP.
- [ ] VMware vSphere v7.0 Update 2 o posterior.
- [ ] vCenter 7.0 Update 2 o posterior.
- [ ] VM version 15 o posterior.
- [ ] No controladores vSpehre CSI third party en el cluster.
- [ ] Definir permisos para la cuenta de servicio de vCenter.
- [ ] Crear recursos necesarios para la instalación de OCP en vCenter.
- [ ] Configurar DNS para la instancia de vCenter que contendrá el cluster de OCP.
- [x] Definir cantidad de nodos y recursos para el cluster.
- [ ] Implementar método de aprovación de los cluster signing requests (CSRs).
- [x] Configurar DHCP para servir IPs y hostnames a los nodos del cluster.
- [ ] Validar direcciones MAC de las interfaces ethernet de cada VM que cumplan con el rango definido por VMware Organizationally Unique Identifier (OUI).
- [x] Configurar Load Balancer, su API e Ingress.
- [x] Validar configuración DNS.

Descargas

- **Coreos Installer**
- **Openshift V4 Arch x86_64**
- **Openshift V4 Otras Arch**
- **Openshift en vSphere/VMWare UPI Installation Required Files**
- **Openshift Any Platform UPI Installation Required Files**

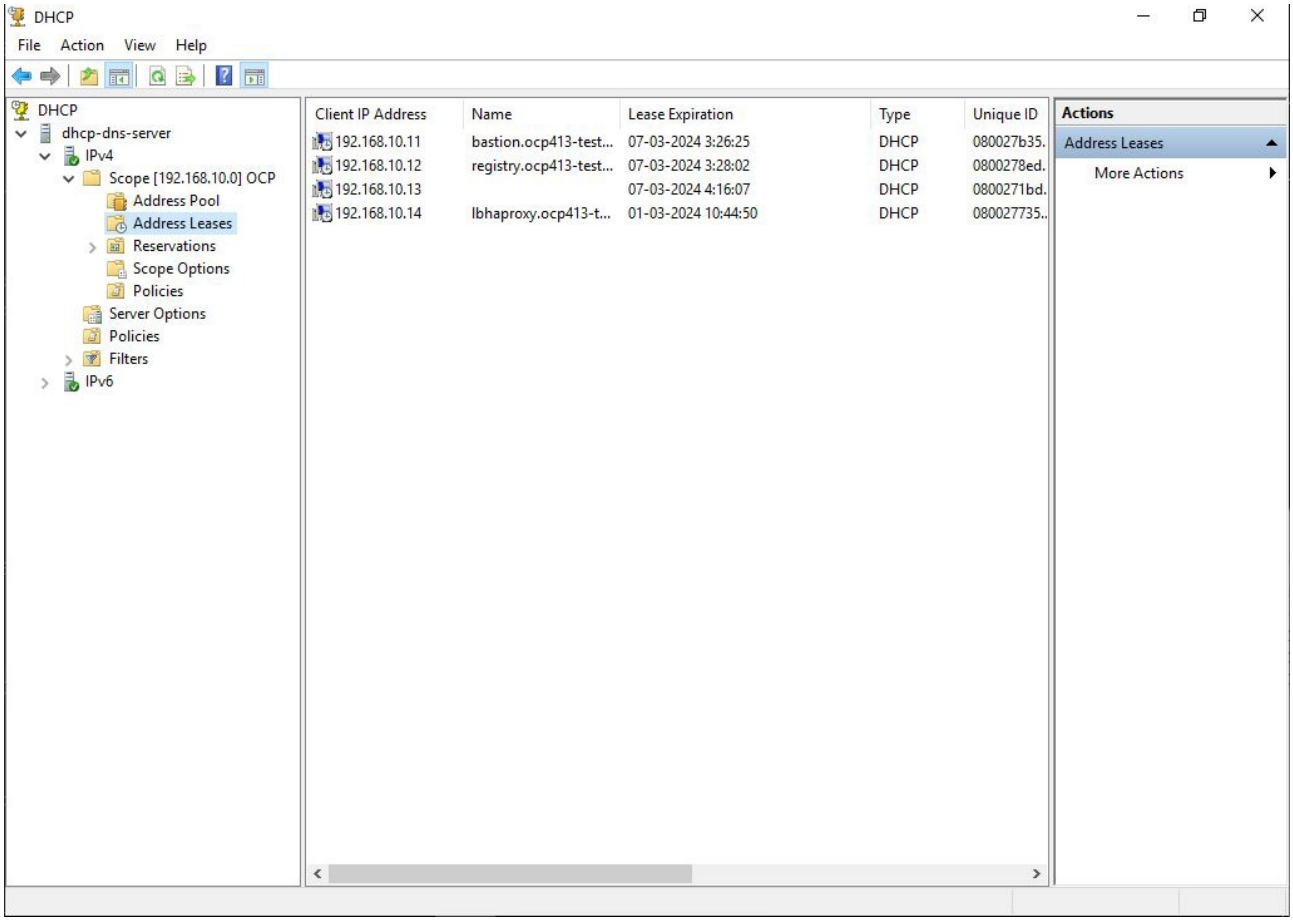
OBS: Si no se logran visualizar las imágenes, desactivar VPN.

1. Máquinas

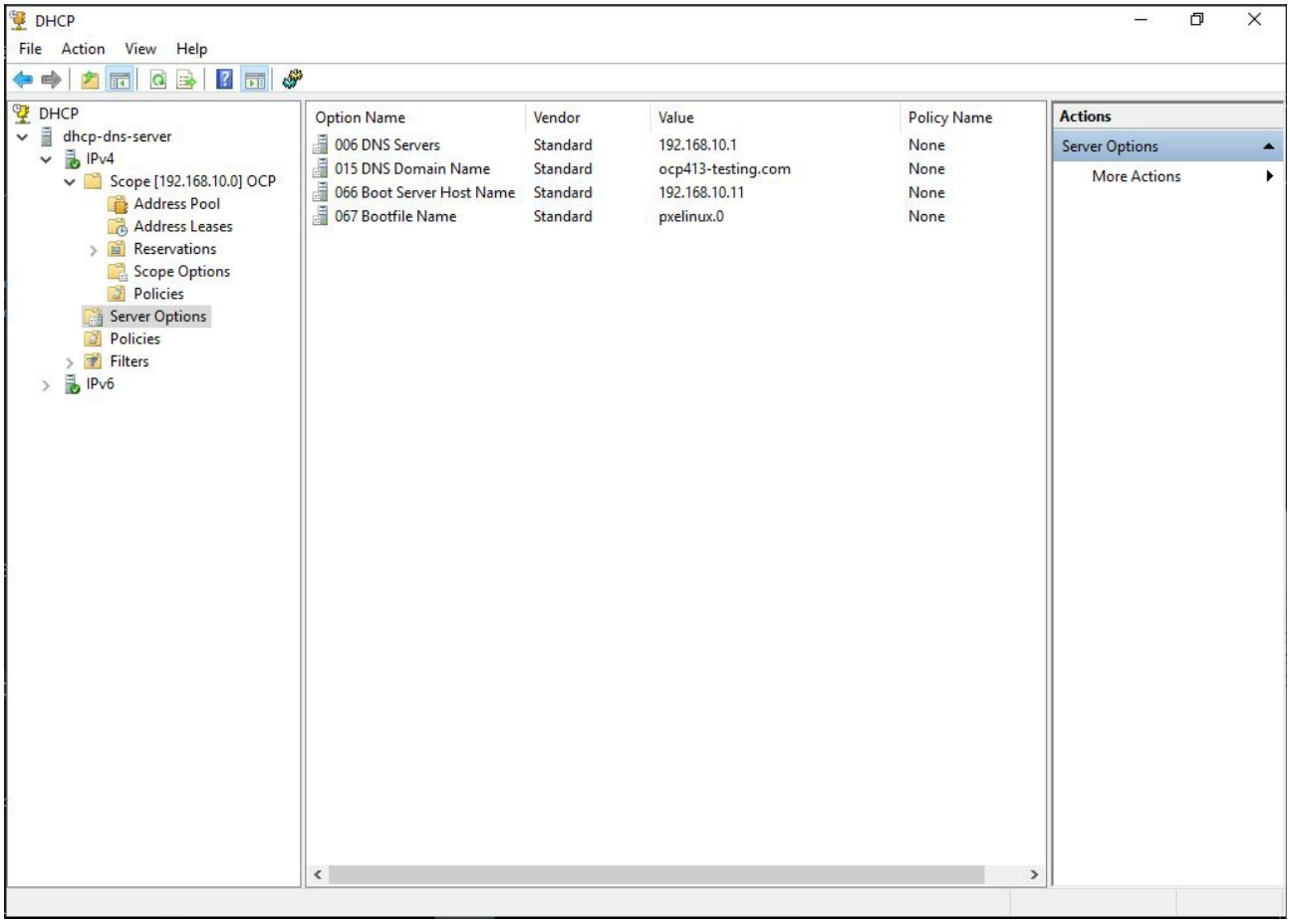
1. DHCP Y DNS

1.1. Windows Server 2016

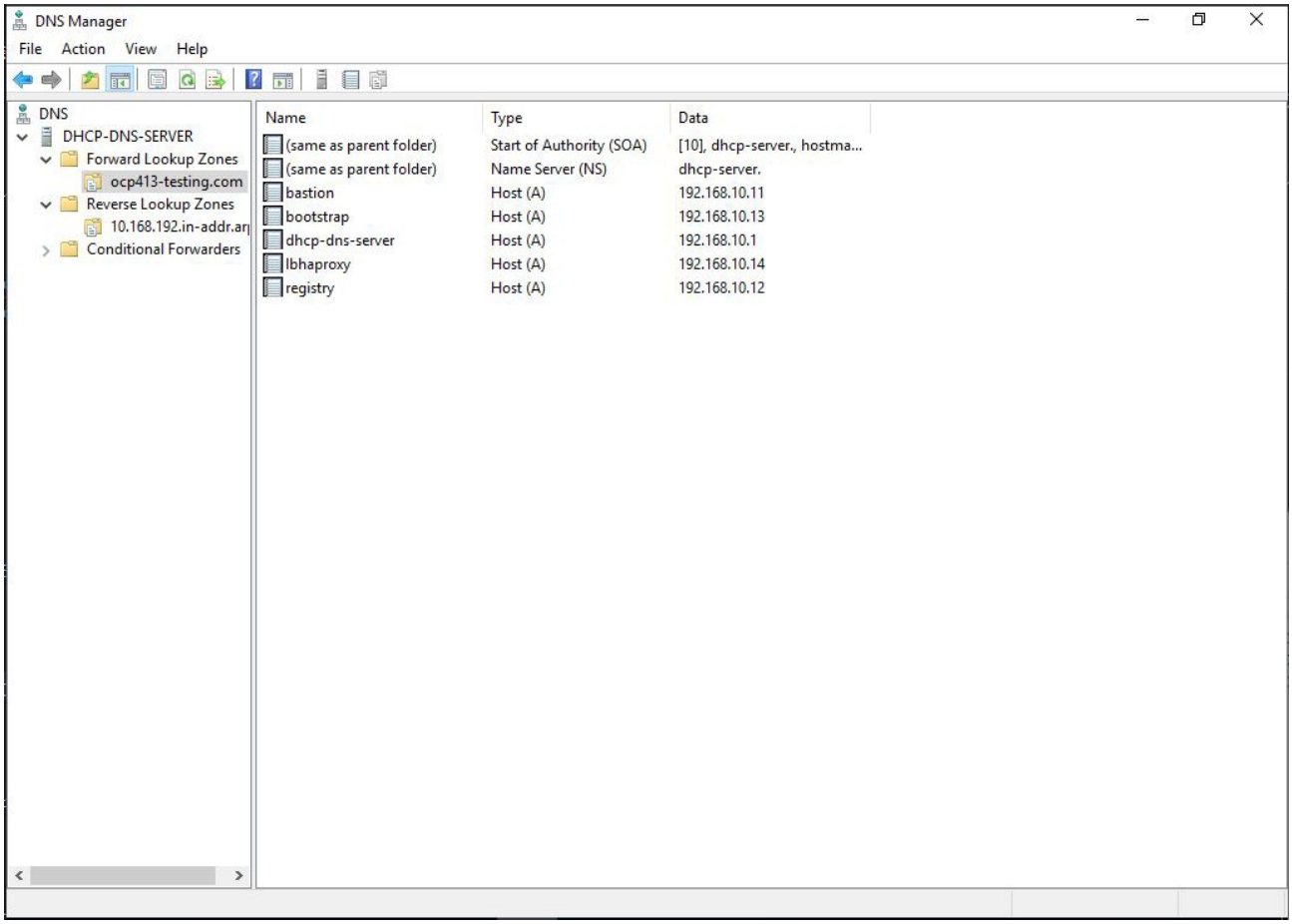
Configuración DHCP

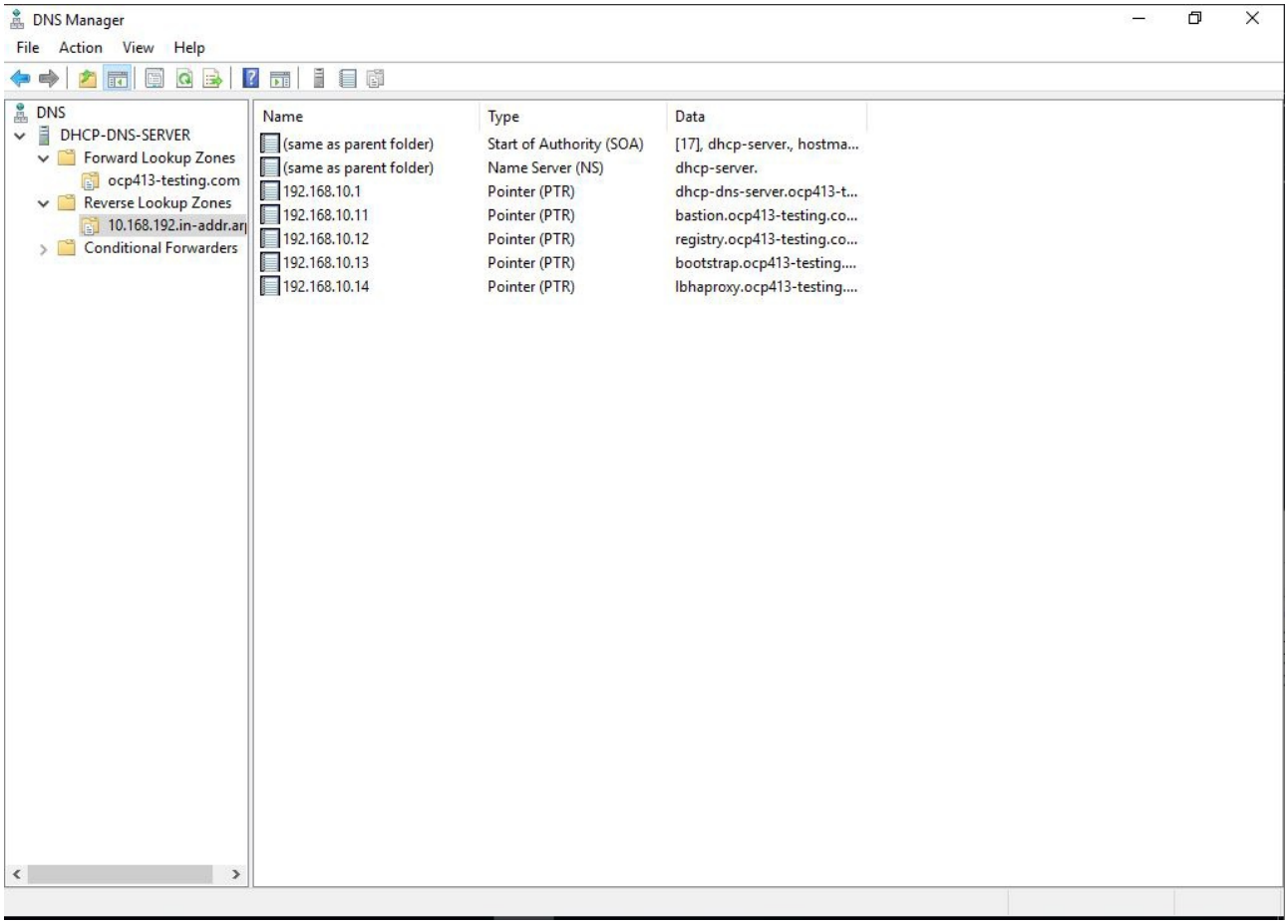


DHCP Server Options



Configuración DNS





1.2. Linux

Instalación de paquetes

```
yum install -y bind dhcp nano
```

Configuración DHCP

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#

default-lease-time 3600;
max-lease-time 7200;
authoritative;
allow booting;
allow bootp;

subnet 192.168.51.0 netmask 255.255.255.0 {
 option domain-search "ocp4waiops.entellab.com";
 range 192.168.51.10 192.168.51.50;
 option routers 192.168.51.1;
 option subnet-mask 255.255.255.0;
 option domain-name-servers 192.168.51.10;
}

host registry {
 hardware ethernet 00:50:56:03:00:f0;
 fixed-address 192.168.51.51;
}

host bootstrap {
 hardware ethernet 00:50:56:03:00:f4;
 fixed-address 192.168.51.100;
 next-server 192.168.51.10;
 filename "pxelinux.0";
}
```

- **default-lease-time 3600:** Este parámetro define el tiempo de predeterminado para las direcciones IP asignadas por el servidor DHCP antes de ser renovadas. En este caso, el tiempo es de 3600 segundos (1 hora).
- **max-lease-time 7200:** Este parámetro especifica el tiempo máximo para las direcciones IP para ser renovadas. En este caso, el tiempo máximo es de 7200 segundos (2 horas).
- **authoritative:** Indica que este servidor DHCP es el servidor oficial para la configuración de direcciones IP en la red especificada.
- **allow booting:** Permite que las máquinas cliente realicen el booteo a través del servidor DHCP.
- **allow bootp:** Permite que las máquinas cliente que utilizan el protocolo BOOTP (Bootstrap Protocol) obtengan configuraciones de red del servidor DHCP.
- **subnet 192.168.51.0 netmask 255.255.255.0:** Se define la configuración para una subred específica.
    - **option domain-search "cp4waiops.entellab.com":** Dominio de búsqueda que los clientes DHCP utilizarán para resolver hostnames.
    - **range 192.168.51.10 192.168.51.50:** Rango de direcciones IP que el servidor DHCP puede asignar a los clientes en esta subred.
    - **option routers 192.168.51.1:** Dirección IP del default gateway para las máquinas cliente en esta subred.
    - **option subnet-mask 255.255.255.0:** Mask de subred que las máquinas cliente deben utilizar para esta subred.
    - **option domain-name-servers 192.168.51.10:** IP del servidor DNS que las máquinas cliente deben utilizar.
    - **host registry :** Esta sección especifica que se reserva una IP para una máquina cliente.
        - **hardware ethernet 00:50:56:03:00:f0:** MAC de la máquina cliente.
        - **fixed-address 192.168.51.51:** IP éstatica para la máquina cliente. Ésta debe estar fuera del rango de subred especificado anteriormente.
    - **host bootstrap :** Similar al caso anterior, solo que ésta configuración tiene párametros adicionales para un booteo por red (PXE).
        - **next-server 192.168.51.10:** IP del servidor TFTP (Trivial File Transfer Protocol) al que la máquina cliente debe comunicarse para obtener el archivo de arranque.
        - **filename "pxelinux.0":** Nombre del archivo de booteo que la máquina cliente debe cargar desde el servidor TFTP.

Configuración DNS

Archivo "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 { 192.168.51.10; };
//      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";
        forward first;
        forwarders { 8.8.8.8; };
        allow-query     { any; };

        /*
         - 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 parts 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";
};

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

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

//FORWARD
zone "entellab.com" IN {
        type master;
        file "forward.entellab.com";
        allow-update { none; };
};

//REVERSE
zone "51.168.192.in-addr.arpa" IN {
```

```
        type master;
        file "reverse.entellab.com";
        allow-update { none; };
};

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

- **listen-on port 53 { 192.168.51.10; }:** IP en la que el servidor DNS escuchará las consultas hacia éste. El puerto 53 es el default para el tráfico de DNS.
- **directory "/var/named":** Directorio en donde el se guardan los archivos de zona del servidor DNS.
- **dump-file "/var/named/data/cache_dump.db":** Archivo en donde se guardará el cache del servidor DNS.
- **statistics-file "/var/named/data/named_stats.txt":** Archivo de estadisticas del servidor DNS.
- **recursing-file "/var/named/data/named.recursing":** Archivo de consultas recursivas al servidor DNS.
- **forward first;:** Opción que indica que se debe buscar recursivamente primero antes de buscar en zonas locales.
- **forwarders { 8.8.8.8; }:** Servidores DNS a los que se deben reenviar las consultas que el servidor DNS no puede resolver localmente. En este caso, se utiliza el servidor DNS de Google (8.8.8.8).
- **allow-query { any; }:** Define quién puede realizar consultas DNS al servidor. En este caso, se permite a cualquier cliente realizar consultas.
- **dnssec-enable yes; y dnssec-validation yes;:** Habilitan la validación de DNSSEC (Domain Name System Security Extensions) para este servidor DNS.
- **bindkeys-file "/etc/named.root.key";:** Especifica la ubicación del archivo que contiene las trust keys necesarias para la resolución de DNSSEC.
- **managed-keys-directory "/var/named/dynamic";:** Directorio donde se almacenarán las trust keys descargadas automáticamente.
- **zone "." IN ;:** Zona raíz.
    - **file "named.ca";:** Archivo que contiene información sobre los servidores raíz de DNS en Internet.
- **zone "cp4waiops.entellab.com" IN ;:** Definición zona FORWARD.
    - **file "forward.entellab.com";:** Archivo de configuración de la zona.
    - **allow-update { none; };:** Indica que no se permiten actualizaciones dinamicas en esta zona.
- **zone "10.168.192.in-addr.arpa" IN ;:** Definición de zona REVERSE.

Archivo Forward.

```
$TTL 3H
@       IN SOA   dns.entellab.com. root.entellab.com. (
                                      10       ; serial
                                      1D      ; refresh
                                      1H      ; retry
                                      1W      ; expire
                                      3H )    ; minimum


@                                     IN      NS      dns.entellab.com.
dns.entellab.com.                     IN      A       192.168.51.10
registry.ocp4waiops.entellab.com.     IN      A       192.168.51.51

bastion.ocp4waiops.entellab.com.      IN      CNAME   dns.entellab.com.
lb.ocp4waiops.entellab.com.           IN      CNAME   dns.entellab.com.

;OCP CLUSTER
bootstrap.ocp4waiops.entellab.com.    IN      A       192.168.51.100

master1.ocp4waiops.entellab.com.      IN      A       192.168.51.101
master2.ocp4waiops.entellab.com.      IN      A       192.168.51.102
master3.ocp4waiops.entellab.com.      IN      A       192.168.51.103

worker1.ocp4waiops.entellab.com.      IN      A       192.168.51.110
worker2.ocp4waiops.entellab.com.      IN      A       192.168.51.111

;infra1.ocp4waiops.entellab.com.       IN      A       192.168.51.130
;infra2.ocp4waiops.entellab.com.       IN      A       192.168.51.131
;infra3.ocp4waiops.entellab.com.       IN      A       192.168.51.132


api.ocp4waiops.entellab.com.          IN      CNAME   lb.ocp4waiops.entellab.com.
api-int.ocp4waiops.entellab.com.      IN      CNAME   lb.ocp4waiops.entellab.com.

apps                                  IN      CNAME   lb.ocp4waiops.entellab.com.
*.apps.ocp4waiops.entellab.com.       IN      CNAME   lb.ocp4waiops.entellab.com.
```

Archivo Reverse.

```
$TTL 3H
@       IN SOA    dns.entellab.com. root.entellab.com. (
                                      10       ; serial
                                      1D      ; refresh
                                      1H      ; retry
                                      1W      ; expire
                                      3H )    ; minimum

@       IN      NS      dns.entellab.com.

10.51.168.192.in-addr.arpa.     IN      PTR     bastion.ocp4waiops.entellab.com.
10.51.168.192.in-addr.arpa.     IN      PTR     dns.entellab.com.
10.51.168.192.in-addr.arpa.     IN      PTR     pxe.ocp4waiops.entellab.com.
10.51.168.192.in-addr.arpa.     IN      PTR     lb.ocp4waiops.entellab.com.
51.51.168.192.in-addr.arpa.     IN      PTR     registry.ocp4waiops.entellab.com.
100.51.168.192.in-addr.arpa.     IN      PTR     bootstrap.ocp4waiops.entellab.com.
101.51.168.192.in-addr.arpa.     IN      PTR     master1.ocp4waiops.entellab.com.
102.51.168.192.in-addr.arpa.     IN      PTR     master2.ocp4waiops.entellab.com.
103.51.168.192.in-addr.arpa.     IN      PTR     master3.ocp4waiops.entellab.com.
110.51.168.192.in-addr.arpa.     IN      PTR     worker1.ocp4waiops.entellab.com.
111.51.168.192.in-addr.arpa.     IN      PTR     worker2.ocp4waiops.entellab.com.
```

- Evidencias de configuración DNS.

```
   dig +noall +answer @<nameserver_ip> api.<cluster_name>.<base_domain>
   dig +noall +answer @<nameserver_ip> api-int.<cluster_name>.<base_domain>
   dig +noall +answer @<nameserver_ip> random.apps.<cluster_name>.<base_domain>
   dig +noall +answer @<nameserver_ip> console-openshift-console.apps.<cluster_name>.<base_domain>
   dig +noall +answer @<nameserver_ip> bootstrap.<cluster_name>.<base_domain>
   dig +noall +answer @<nameserver_ip> -x 192.168.1.5
   dig +noall +answer @<nameserver_ip> -x 192.168.1.96
```

```
dig +noall +answer @192.168.51.10 api.ocp4waiops.entellab.com
dig +noall +answer @192.168.51.10 api-int.ocp4waiops.entellab.com
dig +noall +answer @192.168.51.10 random.apps.ocp4waiops.entellab.com
dig +noall +answer @192.168.51.10 console-openshift-console.apps.ocp4waiops.entellab.com
dig +noall +answer @192.168.51.10 bootstrap.ocp4waiops.entellab.com
dig +noall +answer @192.168.51.10 -x 192.168.51.10
dig +noall +answer @192.168.51.10 -x 192.168.51.99
```

```
[root@bastion ~]# dig +noall +answer @192.168.51.10 api.ocp4waiops.entellab.com
api.ocp4waiops.entellab.com. 10800 IN   CNAME   lb.ocp4waiops.entellab.com.
lb.ocp4waiops.entellab.com. 10800 IN    CNAME   dns.entellab.com.
dns.entellab.com.       10800   IN      A       192.168.51.10
[root@bastion ~]# dig +noall +answer @192.168.51.10 api-int.ocp4waiops.entellab.com
api-int.ocp4waiops.entellab.com. 10800 IN CNAME lb.ocp4waiops.entellab.com.
lb.ocp4waiops.entellab.com. 10800 IN    CNAME   dns.entellab.com.
dns.entellab.com.       10800   IN      A       192.168.51.10
[root@bastion ~]# dig +noall +answer @192.168.51.10 random.apps.ocp4waiops.entellab.com
random.apps.ocp4waiops.entellab.com. 10800 IN CNAME lb.ocp4waiops.entellab.com.
lb.ocp4waiops.entellab.com. 10800 IN    CNAME   dns.entellab.com.
dns.entellab.com.       10800   IN      A       192.168.51.10
[root@bastion ~]# dig +noall +answer @192.168.51.10 console-openshift-console.apps.ocp4waiops.entellab.com
console-openshift-console.apps.ocp4waiops.entellab.com. 10800 IN CNAME lb.ocp4waiops.entellab.com.
lb.ocp4waiops.entellab.com. 10800 IN    CNAME   dns.entellab.com.
dns.entellab.com.       10800   IN      A       192.168.51.10
[root@bastion ~]# dig +noall +answer @192.168.51.10 bootstrap.ocp4waiops.entellab.com
bootstrap.ocp4waiops.entellab.com. 10800 IN A   192.168.51.99
[root@bastion ~]# dig +noall +answer @192.168.51.10 -x 192.168.51.10
10.51.168.192.in-addr.arpa. 10800 IN    PTR     lb.ocp4waiops.entellab.com.
10.51.168.192.in-addr.arpa. 10800 IN    PTR     bastion.ocp4waiops.entellab.com.
10.51.168.192.in-addr.arpa. 10800 IN    PTR     pxe.ocp4waiops.entellab.com.
10.51.168.192.in-addr.arpa. 10800 IN    PTR     dns.entellab.com.
[root@bastion ~]# dig +noall +answer @192.168.51.10 -x 192.168.51.99
99.51.168.192.in-addr.arpa. 10800 IN    PTR     bootstrap.ocp4waiops.entellab.com.
[root@bastion ~]#
```

2. Registry

Para la configuración del registry se deberá disponer de 2 máquinas:

- Una máquina que será la que contenga el "registry container". Lo ideal es usar un OS como CentOS o RHEL. No importa demasiado, pero es recomendable un OS de Redhat. Em este ejemplo se utilizó una VM CentOS 7.
- Una máquina que tenga acceso a internet y a la red interna. Puede ser cualquier máquina, ya sea una VM con cualquier OS o el equipo personal. Lo importante es que tenga acceso a internet y a la red interna. En este ejemplo se utilizó una VM Ubuntu.

Hay 2 formas de configurar e instalar el contenedor Registry.

- Que el contenedor registry esté totalmente aislado de internet y que el servidor Bastión (o el servidor que tiene acceso a internet para el mirroring de imágenes) no pueda acceder al contenedor registry.
- Que el contenedor registry sea accesible por el servidor Bastión (o el servidor que tiene acceso a internet para el mirroring de imágenes) y que esté aislado de internet.

OBS: Para ambos casos el servidor Registry en una primera instancia deberá tener acceso a internet para poder descargar la imágen "registry:2" de docker, necesaria para realizar el mirroring de imágenes y las dependencias necesarias. Una vez terminada la creación del contenedor, se podrá aislar la máquina del registry.

Primer método: Registry pertenece a red interna.

1. Creación del contenedor registry

Instalación de podman

```
yum install -y podman
```

Instalar httpd-tools

```
yum install -y httpd-tools
```

Instalar nano, si es que no lo está. (Opcional)

```
yum install -y nano
```

Crear directorios para el registry

```
mkdir -p /opt/registry/auth
mkdir -p /opt/registry/certs
mkdir -p /opt/registry/data
```

- El directorio /opt/registry/auth es donde se almacenarán los usuarios y contraseñas para el acceso al registry.
- El directorio /opt/registry/certs es donde se almacenarán los certificados para la autenticación del registry.
- El directorio /opt/registry/data es donde se almacenarán las imágenes del registry.

Generar credenciales para el acceso al registry.

```
htpasswd -bBc /opt/registry/auth/htpasswd <username> <password>
```

- **-b**: Proporciona la contraseña en la línea de comandos.
- **-B**: Utiliza el algoritmo bcrypt para cifrar la contraseña.
- **-c**: Crea un nuevo archivo de contraseñas. Si el archivo ya existe, lo sobrescribe.

- **username**: Nombre de usuario para el acceso al registry.
- **password**: Contraseña del usuario para el acceso al registry.

Esto generará un archivo con el nombre de usuario y una contraseña en base64 para el acceso al registry como se muestra a continuación:

```
[root@registry auth]# ls -lrt
total 4
-rw-r--r--. 1 root root 68 feb  8 12:16 htpasswd
[root@registry auth]# cat htpasswd
jotape:$2y$05$K.xsRaIOXd8vTSwF553Lxe8fCotL0NROkHZaYcalVEuTzkGSkXqNq
```

Creación del certificado TLS SAN (Subject Alternative Name). En este ejemplo se utilizará un certificado self-signed.

Fuente

1. Ingresar al directorio */opt/registry/certs*.

2. Generar una llave privada

```
openssl genrsa -des3 -out registry.key 2048
```

- **genrsa**: Genera una llave RSA.
- **-des3**: Cifra la llave con el algoritmo DES3.
- **-out**: Especifica el nombre del archivo de salida.
- **2048**: Especifica el tamaño de la llave en bits.

3. Generar CSR (Certificate Signing Request)

```
openssl req -new -key registry.key -out registry.csr
```

- **req**: Utiliza el comando de solicitud de certificado.
- **-new**: Crea una nueva solicitud de certificado CSR.
- **-key**: Especifica la llave privada. (registry.key).
- **-out**: Especifica el nombre del archivo de salida. (registry.csr).

Cuando pida ingresar el "Common Name", ingresar el hostname de la máquina. En este ejemplo se utlizó el CN = registry

4. Remover Passphrase de la llave

```
cp registry.key registry.key.org
openssl rsa -in registry.key.org -out registry.key
```

5. Crear un config file para SAN

```
touch v3.ext
```

- Contenido del archivo:

```
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints       = CA:TRUE
keyUsage               = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement,
keyCertSign
subjectAltName         = DNS:registry, DNS:registry.ocp4waiops.entellab.com
issuerAltName          = issuer:copy
```

Aqui se define como el servidor DNS resuelve a la máquina que contiene el registry. En este caso se le asignan 2 valores, el hostname de la

máquina para acceder al contenedor de manera local y el nombre de dominio que le asigna el servidor DNS para que otrás máquinas dentro de la red puedan acceder al registry.

6. Generar el certificado self-signed

```
openssl x509 -req -in registry.csr -signkey registry.key -out registry.crt -days 365 -sha256 -extfile v3.ext
```

- **x509**: Utiliza el comando de certificado x509. x509 es un certificado digital estándar.
- **-req**: Se indica que se está trabajando con una solicitud de certificado CSR como input.
- **-in**: Especifica el archivo de entrada que contiene la solicitud de firma de certificado (CSR) que se quiere firmar. (registry.csr).
- **-signkey**: Especifica el archivo que contiene la llave privada con el que se firmára la solicitud del certificado. (registry.key).
- **-out**: Especifica el nombre del archivo de salida. (registry.crt).
- **-days**: Especifica la cantidad de días que el certificado será válido. (1 año).
- **-sha256**: Especifica el algoritmo de hash que se utilizará para firmar el certificado.
- **-extfile**: Especifica el archivo de configuración que contiene el Subject Alternative Name (SAN). (v3.ext).

7. Añadir certificado a los trusted certificates del sistema (CentOS).

Este paso hay que realizarlo en todas lás máquinas que deben acceder al contenedor registry.

```
cp /opt/registry/certs/registry.crt /etc/pki/ca-trust/source/anchors/

update-ca-trust

trust list | grep -i "registry"
```

8. Iniciar el registry

```
podman run --name registry -p 5000:5000 -v /opt/registry/data:/var/lib/registry:z -v /opt/registry/auth:/auth:z -e
"REGISTRY_AUTH=htpasswd" -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" -e
"REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd" -v /opt/registry/certs:/certs:z -e
"REGISTRY_HTTP_TLS_CERTIFICATE=/certs/registry.crt" -e "REGISTRY_HTTP_TLS_KEY=/certs/registry.key" -e
REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true -d registry:2
```

- **run**: Crea y ejecuta un contenedor.
- **--name**: Nombre del contenedor. (registry).
- **-p**: Mapea el puerto del contenedor al puerto del host. El puerto 5000 es un estandar de este tipo de contenedores. (5000:5000).
- **-v /opt/registry/data:/var/lib/registry:z**: Monta un volumen en el contenedor. Esto permite almacenar los datos del registry de Docker en el directorio "/opt/registry/data" del host, que estará disponible en el contenedor en "/var/lib/registry". El modificador "registry:z" indica que se debe establecer el contexto de seguridad SELinux para el volumen.
- **-v /opt/registry/auth:/auth:z**: Monta otro volumen para almacenar la autenticación del registry. Esto permite almacenar los archivos de autenticación en el directorio "/opt/registry/auth" del host, que estará disponible en el contenedor en "/auth".
- **-e "REGISTRY_AUTH=htpasswd"**: Establece una variable de entorno dentro del contenedor llamada "REGISTRY_AUTH" con el valor "htpasswd", lo que indica que se utilizará la autenticación basada en htpasswd.
- **-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm"**: Se indica que las credenciales proporcionadas para acceder al registry de Docker deben aplicarse al "reino" llamado "Registry Realm". Esto significa que cuando un usuario intente acceder al registry, se le solicitarán las credenciales para el "reino" especificado, que en este caso es "Registry Realm".
- **-e "REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd"**: Especifica la ruta dentro del contenedor donde se encuentra el archivo htpasswd para la autenticación.
- **-v /opt/registry/certs:/certs:z**: Monta un volumen para almacenar los certificados TLS. Esto permite almacenar los certificados en el directorio "/opt/registry/certs" del host, que estará disponible en el contenedor en "/certs".
- **-e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/registry.crt"**: Especifica la ubicación del certificado TLS para el registry.
- **-e "REGISTRY_HTTP_TLS_KEY=/certs/registry.key"**: Especifica la ubicación de la clave TLS para el registry.

- **-e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true**: Proporciona compatibilidad con los manifests de schema1.
- **-d registry:2**: Especifica la imagen del contenedor a ejecutar. En este caso, se utiliza la imagen "registry:2", que es una imagen oficial de Docker para ejecutar un registry de Docker de la versión 2.

9. Habilitar puerto 5000

```
firewall-cmd --add-port=5000/tcp --zone=internal --permanent

firewall-cmd --add-port=5000/tcp --zone=public --permanent

firewall-cmd --reload
```

Con ufw

```
ufw allow 5000/tcp

ufw reload
```

10. Descargar pull secret. Una vez descargado, guardarlo en formato JSON.

- **Link Pull Secret**

```
cat ./pull-secret | jq . > ~/containers/auth.json.
```

```
{
        "auths": {
                "cloud.openshift.com": {
                        "auth":
"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T1dKSlcyW

                },
                "quay.io": {
                        "auth":
"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T1dKSlcyW

                },
                "registry.connect.redhat.com": {
                        "auth":
"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxWXpRd09UU

                },
                "registry.ocp4waiops.entellab.com:5000": {
                        "auth": "am90YXBlOmpwYWwwNTk4"
                },
                "registry.redhat.io": {
                        "auth":
"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxWXpRd09UU

                }
        }
}
```

- Ejecutar los siguientes comandos para agregar los registry automaticamente al auth.json.

Puede ocurrir que este comando sobre escribe el archivo json en vez de añadir los valores. Es mejor realizar un respaldo del auth.json antes de ejecutar estos comandos.

```
podman login registry.redhat.io --authfile ~/containers/auth.json
podman login quay.io --authfile ~/containers/auth.json
podman login registry:5000 --authfile ~/containers/auth.json
```

11. Una vez habilitado el firewall, se puede aislar la máquina (desconectarla de internet, pero que permanezca dentro de la red interna).

12. Verificar acceso al contenedor registry.

- Verificar si el contenedor está corriendo

```
podman ps
```

- Ejecutar
  - Local

```
curl https://registry:5000/v2/_catalog -u "usuario-registry:password"

OUTPUT: {"repositories":[]}
```

  - Exterior

```
curl https://registry.cp4waiops.entellab.com:5000/v2/_catalog -u "usuario-registry:password"

OUTPUT: {"repositories":[]}
```

3. Configuracion Bastión

1. Agregar el certificado generado en la máquina del registry a los trusted certificates del servidor Bastión.

- Ubuntu:

```
cd /usr/local/share/ca-certificates

openssl s_client -connect registry.ocp4waiops.entellab.com:5000 -servername registry
```

- RHEL o CentOS

```
cd /etc/pki/ca-trust/source/anchors

openssl s_client -connect registry.ocp4waiops.entellab.com:5000 -servername registry
```

- Copiar el certificado desde ----BEGIN CERTIFICATE---- hasta ----END CERTIFICATE----- , crear un archivo .crt y copiar el contenido dentro.

```
touch registry.crt

nano registry.crt
```

- Actualizar los trusted certificates para añadir el certificado del registry.
  - Ubuntu

    ```
    update-ca-certificates
    ```

  - RHEL o CentOS

    ```
    update-ca-trust
    ```

2. Descargar pull secret. Una vez descargado, guardarlo en formato JSON.

- **Link Pull Secret**

```
cat ./pull-secret | jq . > ~/containers/auth.json.
```

```
{
        "auths": {
                "cloud.openshift.com": {
                        "auth":
"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T1dKSlcyW

                },
                "quay.io": {
                        "auth":
"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T1dKSlcyW

                },
                "registry.connect.redhat.com": {
                        "auth":
"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxWXpRd09UU

                },
                "registry.ocp4waiops.entellab.com:5000": {
                        "auth": "am90YXBlOmpwYWwwNTk4"
                },
                "registry.redhat.io": {
                        "auth":
"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxWXpRd09UU

                }
        }
}
```

- Ejecutar los siguientes comandos para agregar los registry automaticamente al auth.json.

Puede ocurrir que este comando sobre escribe el archivo json en vez de añadir los valores. Es mejor realizar un respaldo del auth.json antes de ejecutar estos comandos.

```
podman login registry.redhat.io --authfile ~/containers/auth.json
podman login quay.io --authfile ~/containers/auth.json
podman login registry:5000 --authfile ~/containers/auth.json
```

3. En el archivo .bashrc del usuario root, agregar las siguientes variables de entorno.

```
export OCP_RELEASE=4.13.10
export LOCAL_REGISTRY=registry.ocp4waiops.entellab.com:5000
export LOCAL_REPOSITORY=ocp4/openshift4
export PRODUCT_REPO=openshift-release-dev
export RELEASE_NAME=ocp-release
export LOCAL_SECRET_JSON=~/containers/auth.json
export ARCHITECTURE=x86_64
export REMOVABLE_MEDIA_PATH=~/images
```

- **OCP_RELEASE**: Versión de OCP a instalar.
- **LOCAL_REGISTRY**: Dominio del contenedor registry local y puerto de la máquina que contiene el contenedor registry.
- **LOCAL_REPOSITORY**: Nombre que tendrá el repositorio en el registry local.
- **PRODUCT_REPO**: Repositorio de Redhat. Para producción se ocupa openshift-release.
- **RELEASE_NAME**: Nombre del release. Para producción se ocupa ocp-release.
- **LOCAL_SECRET_JSON**: Ruta del archivo auth.json.
- **ARCHITECTURE**: Arquitectura de la máquina. Pueden ser x86_64, aarch64, s390x, or ppc64le.
- **REMOVABLE_MEDIA_PATH**: Ruta donde se almacenarán las imágenes.

4. Una vez configuradas las variables de entorno, ejecutar el siguiente comando.

```
source .bashrc
```

5. Verificado lo anterior, ejecutar el siguiente comando.

- Este comando lleva un parametro "--dry-run" que permite verificar si el comando se ejecutará correctamente y no ejecutará nada.
- OBS: Ejecutar este comando en el servidor registry, para hacer esto, se requiere que el servidor registry tenga instalado el CLI **"oc"**, **"oc-mirror"** y **"kubectl"**. Tambien es necesario que se configuren las mismas variables de entorno que se configuraron en el servidor Bastion, solo si es que el servidor Bastion no tiene acceso al contenedor registry.

```
    oc adm release mirror -a ${LOCAL_SECRET_JSON} --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-
${ARCHITECTURE} --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} --to-release-
image=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-${ARCHITECTURE} --dry-run
```

6. Al terminar la ejecución del comando anterior, entregará un output como el siguiente:

```
    To use the new mirrored repository to install, add the following section to the install-config.yaml:

    imageContentSources:
    - mirrors:
      - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
      source: quay.io/openshift-release-dev/ocp-release
    - mirrors:
      - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
      source: quay.io/openshift-release-dev/ocp-v4.0-art-dev


    To use the new mirrored repository for upgrades, use the following to create an ImageContentSourcePolicy:

    apiVersion: operator.openshift.io/v1alpha1
    kind: ImageContentSourcePolicy
    metadata:
      name: example
    spec:
      repositoryDigestMirrors:
      - mirrors:
        - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
        source: quay.io/openshift-release-dev/ocp-release
      - mirrors:
        - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
        source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

- Es importante guardar este output ya que se utilizará en el archivo de configuración "install-config.yaml" para la instalación del cluster.

7. Mirror de imagenes.

- En este caso se descarguó la version 4.13.10 de OCP. Fueron alrededor de 18 GB de imágenes.

```
    oc adm release mirror -a ${LOCAL_SECRET_JSON} --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-
${ARCHITECTURE} --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} --to-release-
image=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-${ARCHITECTURE}
```

Segundo método: Contenedor registry asilado de red interna.

Este metodo no funcionó. Al parecer hay un bug con el penúltimo comando que hay que ejecutar (oc image mirror). Dejo referencias del posible bug:

- **Referencia de posible bug en comunidad de Redhat 1**
- **Referencia de posible bug en comunidad de Redhat 2**

1. En servidor registry.

1. Descargar pull secret. Una vez descargado, guardarlo en formato JSON.

- **Link Pull Secret**

```
cat ./pull-secret | jq . > ~/containers/auth.json.
```


```
{
        "auths": {
                "cloud.openshift.com": {
                        "auth":
"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T1dKSlcyW

                },
                "quay.io": {
                        "auth":
"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T1dKSlcyW

                },
                "registry.connect.redhat.com": {
                        "auth":
"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxWXpRd09UU

                },
                "registry.ocp4waiops.entellab.com:5000": {
                        "auth": "am90YXBlOmpwYWwwNTk4"
                },
                "registry.redhat.io": {
                        "auth":
"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxWXpRd09UU

                }
        }
}
```


- Ejecutar los siguientes comandos para agregar los registry automaticamente al auth.json.

Puede ocurrir que este comando sobre escribe el archivo json en vez de añadir los valores. Es mejor realizar un respaldo del auth.json antes de ejecutar estos comandos.

```
podman login registry.redhat.io --authfile ~/containers/auth.json
podman login quay.io --authfile ~/containers/auth.json
podman login registry:5000 --authfile ~/containers/auth.json
```

2. En el archivo .bashrc del usuario root, agregar las siguientes variables de entorno.

```
export OCP_RELEASE=4.13.10
export LOCAL_REGISTRY=registry.ocp4waiops.entellab.com:5000
export LOCAL_REPOSITORY=ocp4/openshift4
export PRODUCT_REPO=openshift-release-dev
export RELEASE_NAME=ocp-release
export LOCAL_SECRET_JSON=~/containers/auth.json
export ARCHITECTURE=x86_64
export REMOVABLE_MEDIA_PATH=~/images
```

3. Una vez configuradas las variables de entorno, ejecutar el siguiente comando.

```
source .bashrc
```

2. En servidor Bastion

1. Verificado lo anterior, ejecutar el siguiente comando.

- Este comando lleva un parametro "--dry-run" que permite verificar si el comando se ejecutará correctamente y no ejecutará nada.

```
    oc adm release mirror -a ${LOCAL_SECRET_JSON} --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-
${ARCHITECTURE} --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} --to-release-
image=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-${ARCHITECTURE} --dry-run
```

2. Al terminar la ejecución del comando anterior, entregará un output como el siguiente:

```
    To use the new mirrored repository to install, add the following section to the install-config.yaml:

    imageContentSources:
        - mirrors:
            - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
            source: quay.io/openshift-release-dev/ocp-release
        - mirrors:
            - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
             source: quay.io/openshift-release-dev/ocp-v4.0-art-dev


    To use the new mirrored repository for upgrades, use the following to create an ImageContentSourcePolicy:

    apiVersion: operator.openshift.io/v1alpha1
    kind: ImageContentSourcePolicy
    metadata:
    name: example
    spec:
        repositoryDigestMirrors:
            - mirrors:
                - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
                source: quay.io/openshift-release-dev/ocp-release
            - mirrors:
                - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
                source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

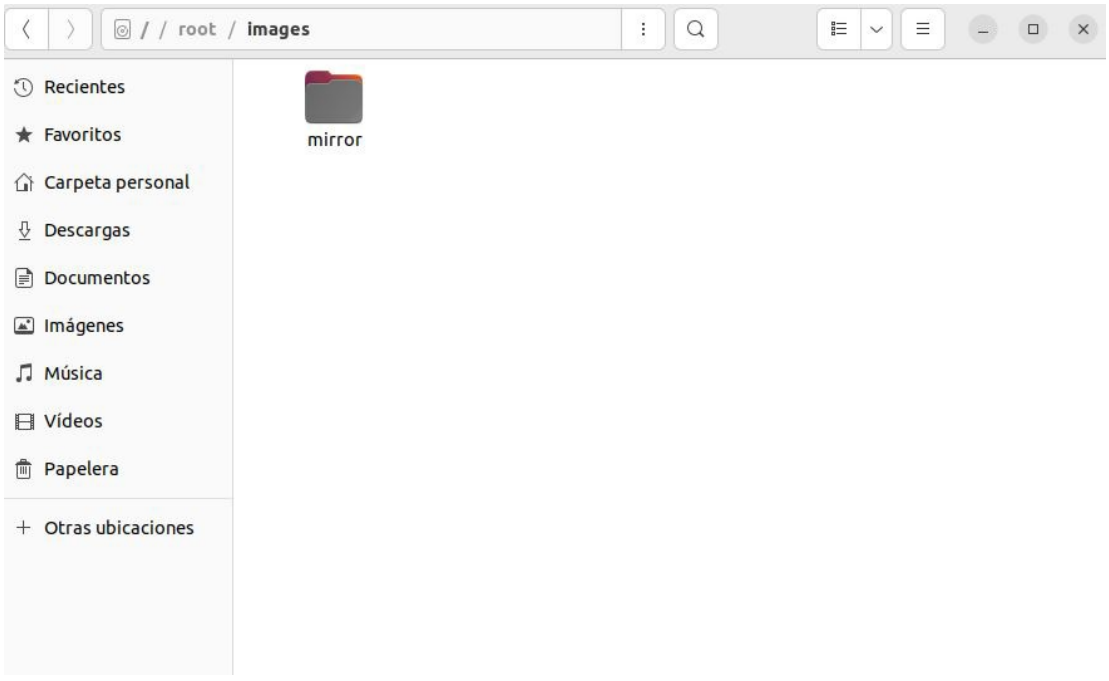
- Es importante guardar este output ya que se utilizará en el archivo de configuración "install-config.yaml" para la instalación del cluster.

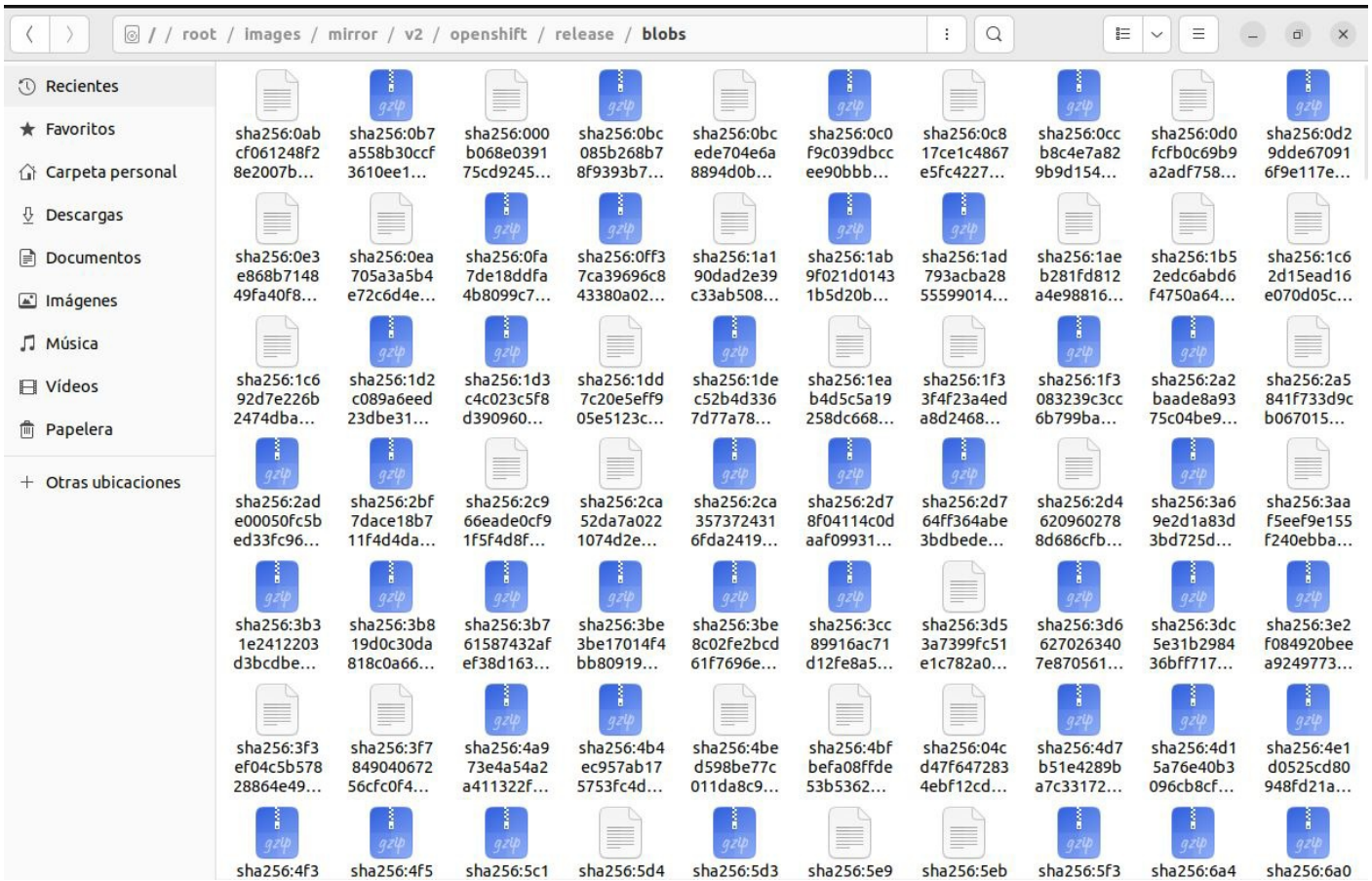
3. Terimnado el comando anterior, ejecutar

```
    oc adm release mirror -a ${LOCAL_SECRET_JSON} --to-dir=${REMOVABLE_MEDIA_PATH}/mirror
quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-${ARCHITECTURE}
```

- Al terminar debería haber una carpeta con el nombre "mirror" en la ruta especificada en la variable de entorno "REMOVABLE_MEDIA_PATH".

```
    ls -l ~/images/mirror
```





4. Una vez terminada la ejecución del comando anterior, traspasar las imágenes al servidor registry.

- OJO: Es necesario habilitar una conexión SSH entre el servidor Bastión y el servidor registry. Esta operación tambien demoró más o menos 30 minutos.

```
scp -r ./images root@<ip-máquina-registry>:~/images
```

5. Al terminar el traspaso de los archivos, ir al servidor registry y hacer un deploy de las imágenes dentro del contenedor creado previamente.

- OBS 1: Se requiere que la máquina registry tenga los CLI "oc" y "kubectl".

- OBS 2: Éste es el comando que se mencionó al principio que genera conflicto. Es el único comando que no se logró ejecutar al realizar este método.

```
    oc image mirror -a ${LOCAL_SECRET_JSON} --from-dir=${REMOVABLE_MEDIA_PATH}/mirror
"file://openshift/release:${OCP_RELEASE}*" ${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} --skip-missing
```

6. Verificar imágenes dentro del registry.

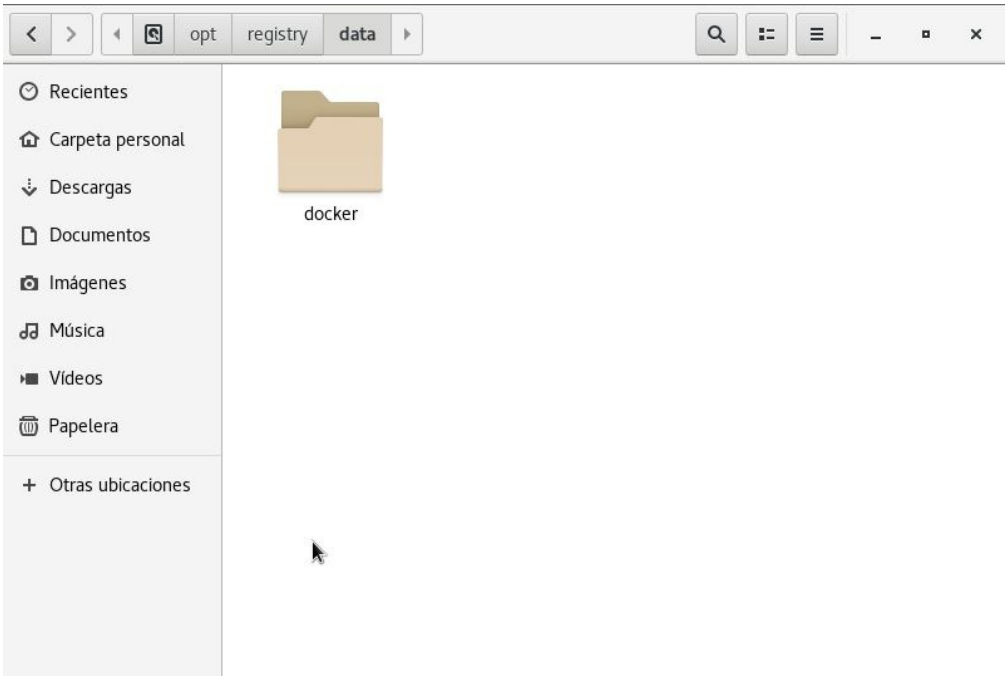
- Una vez finalizado el paso anterior, nos ubicamos en el servidor Registry. Verificar si las imágenes se encuentran en el registry.
  - Local

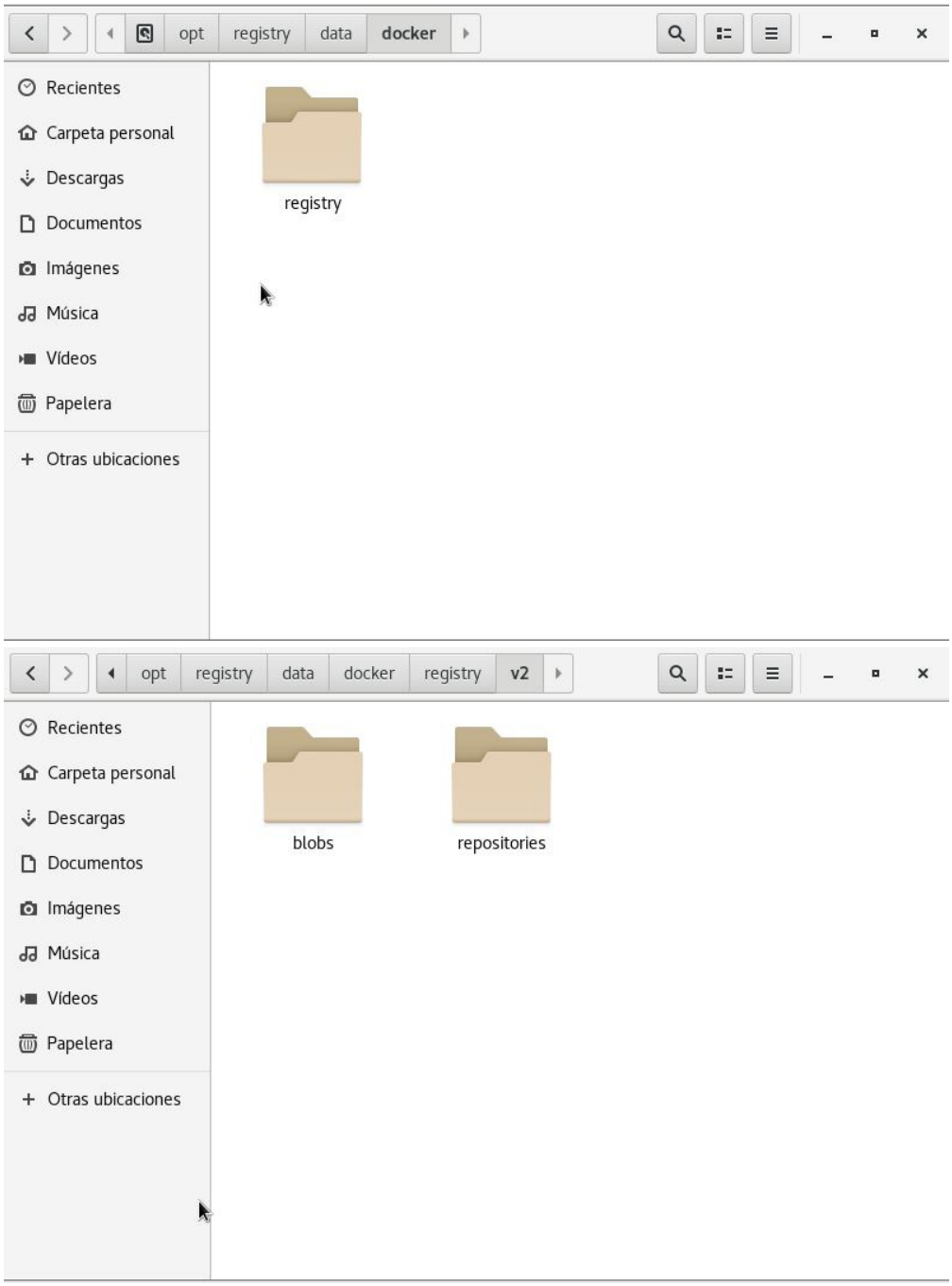
```
curl -u usuario-registry:password registry.ocp4waiops.entellab.com:5000/v2/_catalog

OUTPUT: {"repositories":["ocp4/openshift4"]}
```

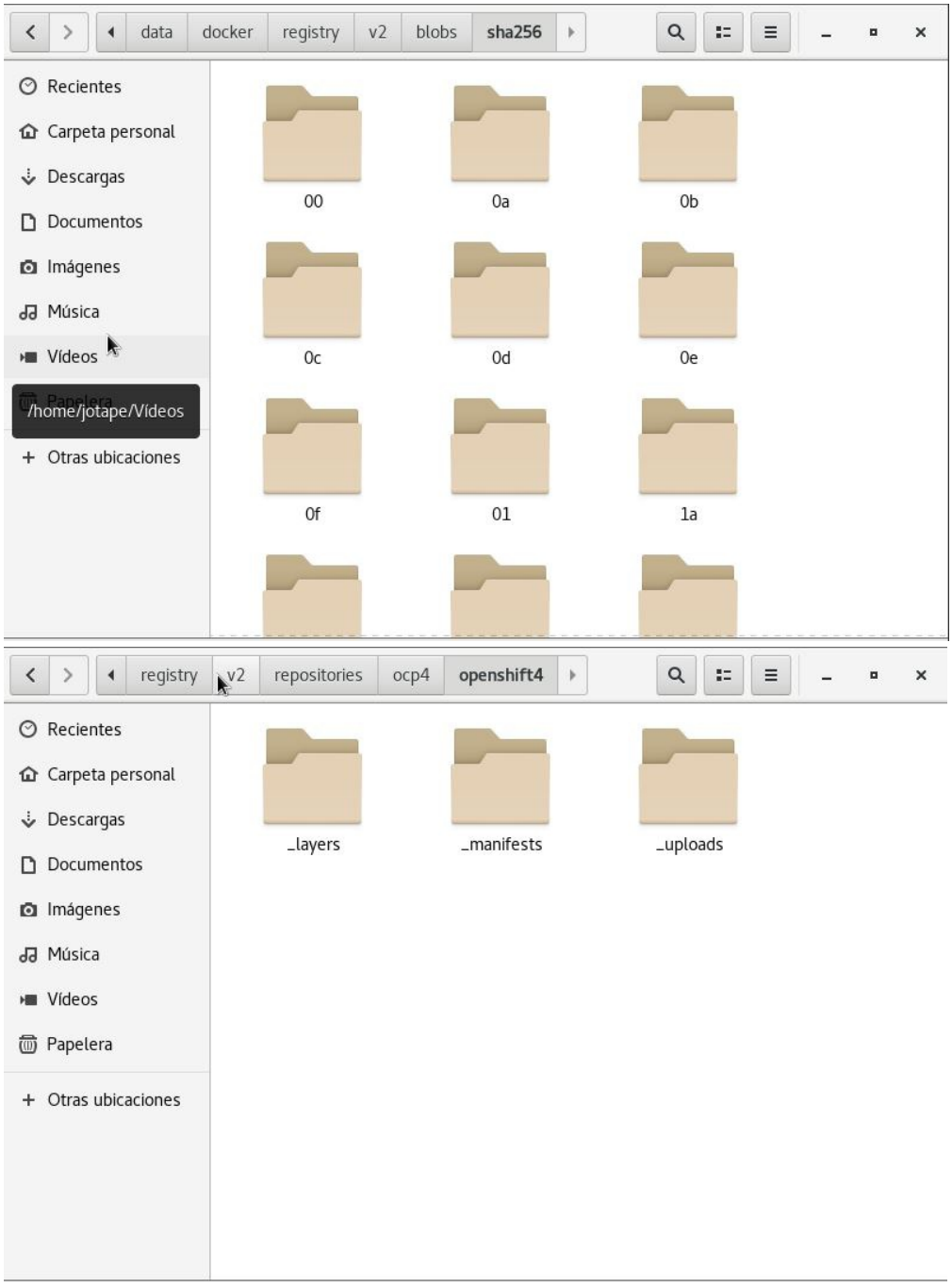
  - Exterior

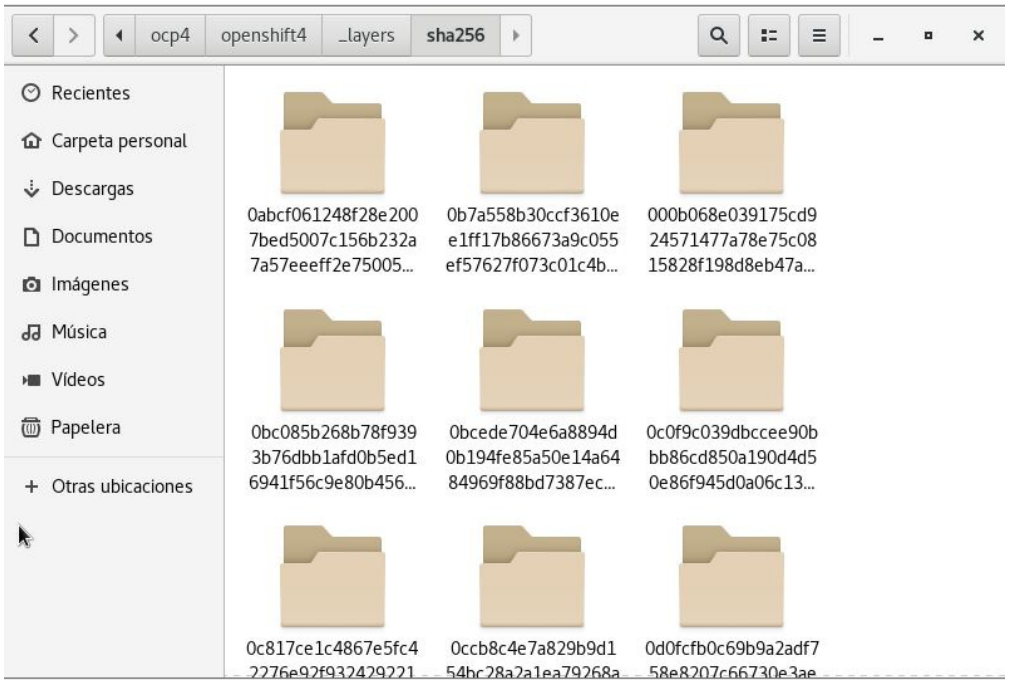
```
curl -u usuario-registry:password registry.cp4waiops.entellab.com:5000/v2/ocp4/openshift4/tags/list

OUTPUT: JSON con las imágenes desplegadas en el contenedor.
```

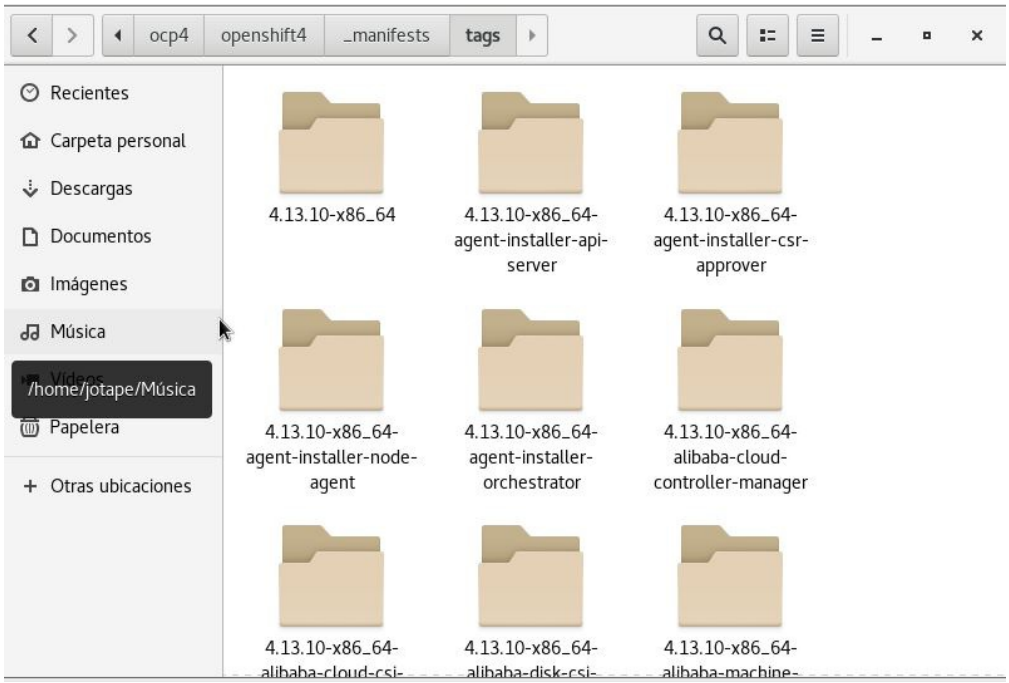
7. Evidencias de archivos en el registry.




opt
registry
data
docker
Recientes
Carpeta personal
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
registry
opt
registry
data
docker
registry
v2
Recientes
Carpeta personal
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
blobs
repositories


data
docker
registry
v2
blobs
sha256
Recientes
Carpeta personal
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
/home/jotape/Vídeos
Otras ubicaciones
00
0a
0b
0c
0d
0e
0f
01
1a
registry
v2
repositories
ocp4
openshift4
Recientes
Carpeta personal
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
_layers
_manifests
_uploads


ocp4
openshift4
_layers
sha256
Recientes
Carpeta personal
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
0abcf061248f28e2007bed5007c156b232a7a57eeeff2e75005...
0b7a558b30ccf3610ee1ff17b86673a9c055ef57627f073c01c4b...
000b068e039175cd924571477a78e75c0815828f198d8eb47a...
0bc085b268b78f9393b76dbb1afd0b5ed16941f56c9e80b456...
0bcede704e6a8894d0b194fe85a50e14a6484969f88bd7387ec...
0c0f9c039dbccee90bbb86cd850a190d4d50e86f945d0a06c13...
0c817ce1c4867e5fc4
0ccb8c4e7a829b9d1
0d0fcfb0c69b9a2adf7
58e8207c66730e3ae


ocp4
openshift4
_manifests
tags
Recientes
Carpeta personal
Descargas
Documentos
Imágenes
Música
/home/jotape/Música
Papelera
Otras ubicaciones
4.13.10-x86_64
4.13.10-x86_64-agent-installer-api-server
4.13.10-x86_64-agent-installer-csr-approver
4.13.10-x86_64-agent-installer-node-agent
4.13.10-x86_64-agent-installer-orchestrator
4.13.10-x86_64-alibaba-cloud-controller-manager
4.13.10-x86_64-alibaba-cloud-csi-
4.13.10-x86_64-alibaba-disk-csi-
4.13.10-x86_64-alibaba-machine-

3. Nodos

1. Bootstrap

1. Configuración

1. Descargar syslinux - tftp-server - http-server - pxelinux

- Ubuntu

```
apt-update
apt install tftpd-hpa apache2 syslinux pxelinux
```

- Iniciar servicios

```
systemctl start tftd-hpa
systemctl enable tftd-hpa
systemctl start apache2
systemctl enable apache2
```

- RHEL o CentOS

```
yum-update
yum install -y tftp-server tftp httpd syslinux
```

  - Iniciar servicios

```
systemctl start tftp-server
systemctl enable tftp-server
systemctl start httpd
systemctl enable httpd
```

2. Obtener manifest e ignition files. Para obtenerlos, primero hay que crear el archivo "install-config.yaml".

- Crear directorio donde alojar el archivo "install-config.yaml", archivos ignition e imágenes para instalar RHCOS.

```
mkdir ~/ignition-files | cd ~/ignition-files

touch install-config.yaml

nano install-config.yaml
```

- install-config.yaml

```
apiVersion: v1
baseDomain: entellab.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: ocp4waiops
imageContentSources:
- mirrors:
  - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - registry.ocp4waiops.entellab.com:5000/ocp4/openshift4
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OVNKubernetes
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: '{"auths":{"cloud.openshift.com":
{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T
{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfOWU3MTlhZWRhMmY1NDRiZjk2Mjc4MTFjNzVmMTNjOGM6Qk9IU1UwVjZFS0VYUVZYUEE2T
{"auth":"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxW
{"auth":"fHVoYy1wb29sLTgyNDJiNzFjLWQzYjgtNDdjMy1iNGNiLWExMDI3ZjAzZmY2NjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSXpNV0UxW
{"auth":"am90YXBlOmpwYWwwNTk4"}}}'
sshKey: 'ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCze5WtXeQnqz28Y/F6r3hRzEqbuU/KKy/e63kn3C546cUPk99Ll7Z6apniBT2IrIfFtamNfnd7AT6wY8PhYVHGIpJqnN
root@bastion'
```

- En el caso de que la instalación se realize en una VMWare y se tenga acceso al vCenter, modificar el siguiente parametro del archivo "install-config.yaml"

```
platform:
  vsphere:
    failureDomains:
    - name: <failure_domain_name>
      region: <default_region_name>
      server: <fully_qualified_domain_name>
      topology:
        computeCluster: "/<datacenter>/host/<cluster>"
        datacenter: <datacenter>
        datastore: "/<datacenter>/datastore/<datastore>"
        networks:
        - <VM_Network_name>
        resourcePool:"/<datacenter>/host/<cluster>/Resources/<resourcePool>"
        folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>"
      zone: <default_zone_name>
    vcenters:
    - datacenters:
      - <datacenter>
      password: <password>
      port: 443
      server: <fully_qualified_domain_name>
      user: administrator@vsphere.local
    diskType: thin
```

Segun documentación:

- **failureDomain**: Establece las relaciones entre una región y una zona. Un dominio de fallos se define mediante objetos vCenter, como un objeto datastore. Un dominio de fallos define la ubicación de vCenter para los nodos de clúster de OpenShift Container Platform.

- **datacenter**: VMWare / vSphere datacenter.
- **datastore**: La ruta al almacén de datos de vSphere que contiene archivos de máquinas virtuales, plantillas e imágenes ISO. IMPORTANTE: Puede especificar la ruta de cualquier almacén de datos que exista en un clúster de almacenes de datos. Por defecto, Storage vMotion se habilita automáticamente para un cluster de datastore. Red Hat no soporta Storage vMotion, por lo que debe desactivar Storage vMotion para evitar problemas de pérdida de datos en su cluster OpenShift Container Platform. Si debe especificar VMs a través de múltiples datastores, utilice un objeto datastore para especificar un dominio de fallo en el archivo de configuración install-config.yaml de su cluster. Para obtener más información, consulte "Habilitación de regiones y zonas de VMware vSphere".
- **resourcePool:** Opcional: Para la infraestructura aprovisionada por el instalador, la ruta absoluta de un pool de recursos existente donde el programa de instalación crea las máquinas virtuales, por ejemplo, /<nombre_centro_de_datos>/host/<nombre_cluster>/Recursos/<nombre_pool_de_recursos>/<nombre_pool_de_recursos_anidado_opcional>. Si no se especifica ningún valor, los recursos se instalan en la raíz del clúster /ejemplo_centro_de_datos/host/ejemplo_cluster/Recursos.
- **folder**: Opcional: Para la infraestructura aprovisionada por el instalador, la ruta absoluta de una carpeta existente en la que el programa de instalación crea las máquinas virtuales, por ejemplo, /<nombre_centro_de_datos>/vm/<nombre_carpeta>/<nombre_subcarpeta>. Si no proporciona este valor, el programa de instalación crea una carpeta de nivel superior en la carpeta de máquinas virtuales del centro de datos que se denomina con el ID de infraestructura. Si está proporcionando la infraestructura para el clúster y no desea utilizar el objeto StorageClass predeterminado, denominado thin, puede omitir el parámetro de carpeta del archivo install-config.yaml.
- **password**: La contraseña asociada al usuario VMWare / vSphere.
- **server**: El nombre de host completo o la dirección IP del servidor vCenter.
- **diskType**: El método de aprovisionamiento de disco de VMWare / vSphere.

3. Ejecutar el siguiente comando para obtener los manifest files.

```
openshift-install create manifests --dir <absolute-path-archivo-install-config.yaml>
```

- Verificar el archivo ***mastersSchedulable*** en la ruta ***<path-archivo-install-config.yaml>/manifests/cluster-scheduler-02-config.yml***
- Cambiar el valor del parametro ***mastersSchedulable*** a **false** si es que no lo está.

4. Ejecutar el siguiente comando para obtener los ignition files. Este comando "consume" el archivo "install-config.yaml". En este ejercicio éste último desapareció, no sé si es correcto que pase eso, si ese es el caso respaldar el contenido del archivo "install-config.yaml" para crearlo nuevamente para cuando se proceda a instalar el cluster.

```
openshift-install create ignition-configs --dir <absolute-path-archivo-install-config.yaml>
```

5. En el mismo directorio donde se encuentra el archivo "install-config.yaml", poner los archivos "bootstrap.ign" y "rhcos-4.13.10-x86_64-live-initramfs.x86_64" para crear una configuración personalizada del PXE Boot.

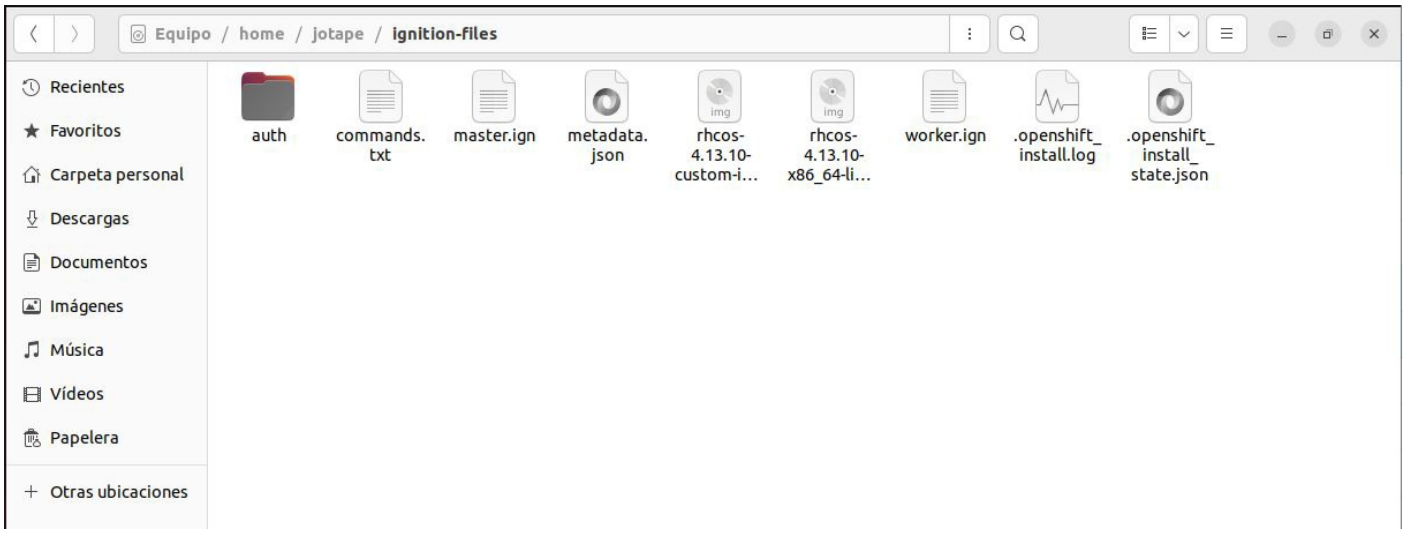
Ejecutar el siguiente comando:

El CLI de coreos-installer no funciona en versiones RHEL o CentOS 7 o anteriores.

```
   coreos-installer pxe customize rhcos-4.13.10-x86_64-live-initramfs.x86_64.img(1) --dest-ignition <path-ign-files>/bootstrap.ign(2) --dest-console tty0(3) --dest-console ttyS0,115200n8(4) --dest-device /dev/sda(5) -o rhcos-4.13.10-x86_64-custom-bootstrap-initramfs.x86_64.img(6)

   coreos-installer pxe customize rhcos-4.13.10-x86_64-live-initramfs.x86_64.img(1) --dest-ignition <path-ign-files>/master.ign(2) --dest-console tty0(3) --dest-console ttyS0,115200n8(4) --dest-device /dev/sda(5) -o rhcos-4.13.10-x86_64-custom-master-initramfs.x86_64.img(6)

   coreos-installer pxe customize rhcos-4.13.10-x86_64-live-initramfs.x86_64.img(1) --dest-ignition <path-ign-files>/worker.ign(2) --dest-console tty0(3) --dest-console ttyS0,115200n8(4) --dest-device /dev/sda(5) -o rhcos-4.13.10-x86_64-custom-worker-initramfs.x86_64.img(6)
```

- (1). Esto invoca el comando coreos-installer, que es una herramienta para instalar CoreOS. pxe customize indica que se está personalizando una imagen para arranque por red (PXE). rhcos-4.13.10-x86_64-live-initramfs.x86_64.img es el nombre de la imagen de inicio de CoreOS que se personalizará.

- (2). Esto especifica el destino de la configuración de Ignition, que es un sistema de inicialización de CoreOS
- (3). Esto indica que la salida de la consola se enviará a la primera consola virtual (tty0).
- (4). Aquí, ttyS0 se refiere al primer puerto serie del sistema. Este parámetro es útil en entornos donde la interacción con el sistema se realiza a través de una conexión serie, como en sistemas embebidos o servidores sin interfaz gráfica. 115200n8 especifica la velocidad de transmisión (baudrate) y la configuración del terminal para el puerto serie.
- (5). Esto especifica el punto de montaje donde se instalará CoreOS.
- (6). Esto indica el nombre del archivo de salida de la imagen personalizada.
- **Referencia de lo realizado anteriormente.**



5. Configurar el TFTP Server

Ubuntu

```
nano /etc/default/tftpd-hpa
```

RHEL o CentOS

```
nano /usr/lib/systemd/system/tftp.service
nano /usr/lib/systemd/system/tftp.socket
```

- Ingresar lo siguiente
  - Ubuntu

    ```
    TFTP_USERNAME="tftp"
    TFTP_DIRECTORY="/var/lib/tftpboot"
    TFTP_ADDRESS="0.0.0.0:69"
    TFTP_OPTIONS="--secure --create"
    ```

  - RHEL o CentOS

    tftp.service

```
[Unit]
Description=Tftp Server
Requires=tftp.socket
Documentation=man:in.tftpd

[Service]
ExecStart=/usr/sbin/in.tftpd -s -p -c  /var/lib/tftpboot
StandardInput=socket

[Install]
WantedBy=multi-user.target
Also=tftp.socket
```

tftp.socket

```
[Unit]
Description=Tftp Server Activation Socket

[Socket]
ListenDatagram=69

[Install]
WantedBy=sockets.target
```

- Guardar y salir.
- Crear directorio */var/lib/tftpboot*

```
mkdir /var/lib/tftpboot
```

- Copiar archivos SYSLINUX dentro del directorio *var/lib/tftpboot*
  - Ubuntu

```
cp /usr/lib/syslinux/modules/bios/* /var/lib/tftpboot/
```

  - RHEL o CentOS

```
cp /usr/share/syslinux/* /var/lib/tftpboot/
```

- Descargar imágen para pantalla de booteo y moverla al directorio /var/lib/tftpboot. (**Opcional**)

```
wget https://raw.githubusercontent.com/leoaaraujo/openshift_pxe_boot_menu/main/files/bg-ocp.png -O
/var/lib/tftpboot/bg-ocp.png
```

- Copiar archivo *pxelinux.0* dentro del directorio */var/lib/tftpboot*.
  - Ubuntu

```
cp /usr/lib/PXELINUX/pxelinux.0 /var/lib/tftpboot/
```

- Mover o copiar los archivos "rhcos-4.13.10-x86*64-live-kernel-x86_64" y "rhcos-4.13.10-x86_64-custom-initramfs.x86_64.img"(bootstrap, master y worker) dentro del directorio */var/lib/tftpboot_

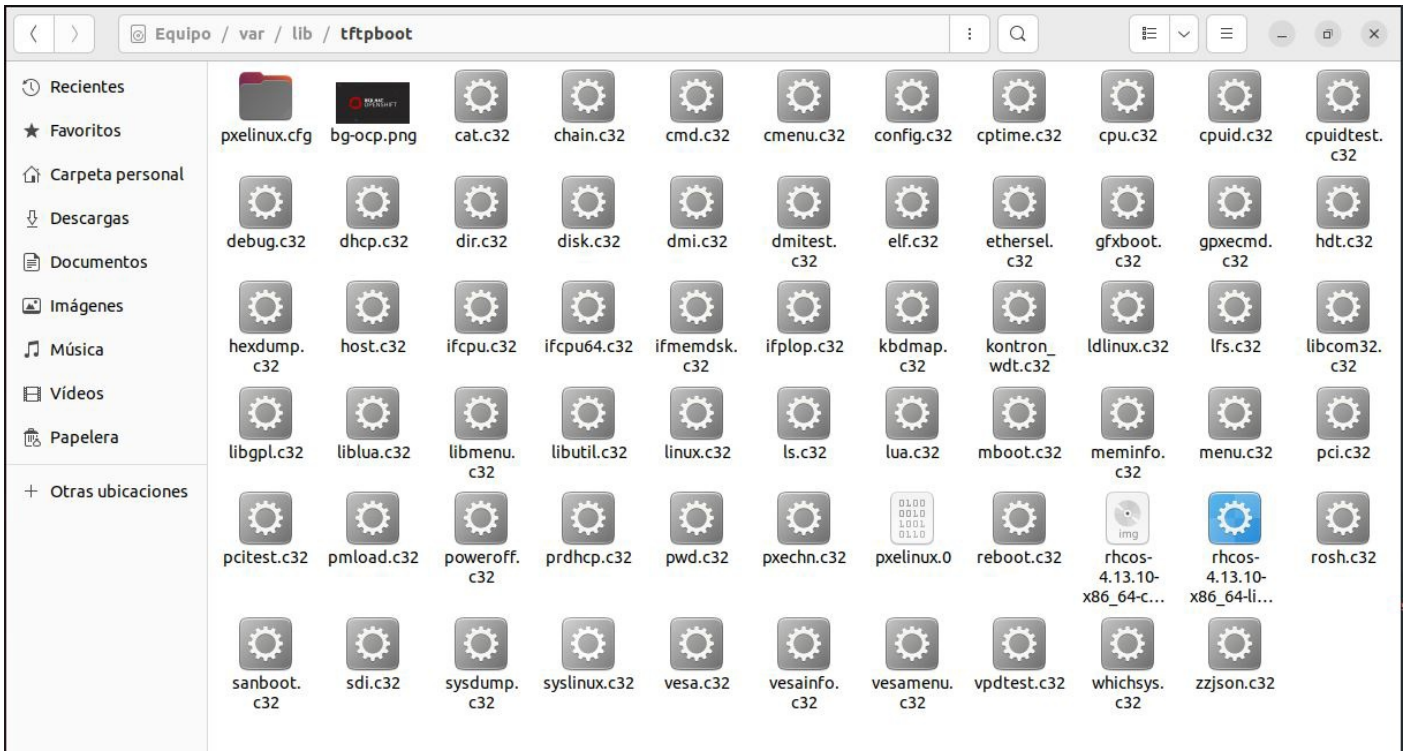
```
mv <path-ignition-files>/rhcos-4.13.10-x86_64-live-kernel-x86_64 /var/lib/tftpboot/

mv <path-ignition-files>/rhcos-4.13.10-x86_64-custom-initramfs.x86_64.img /var/lib/tftpboot/
```

- Crear directorio *pxelinux.cfg* en la ruta */var/lib/tftpboot* y dentro de él crear un archivo *default*.

```
mkdir pxelinux.cfg

cd pxelinux.cfg

touch default
```

- El directorio debería quedar de la siguiente manera:

```
var
|___lib
    |___tftpboot
        |___pxelinux.cfg
            |   |   default
            |   pxelinux.0
            |   bg-ocp.png
            |   rhcos-4.13.10-x86_64-live-kernel-x86_64
            |   rhcos-4.13.10-x86_64-custom-initramfs.x86_64.img
            |   syslinux files...
```



- Quitarle permisos al directorio tftpboot y contenido.

  OBS: Para efecto de este ejercicio se le quitaron todos los permisos las rutas, no es la mejor opción pero aun no está claro qué permisos necesita el booteo PXE para obtener el archivo de arranque.

  - Ubuntu

```
chown -R nobody:nogroup /var/lib/tftpboot
chmod -R 777 /var/lib/tftpboot
```

  - RHEL o CentOS

```
chown -R nobody:nobody /var/lib/tftpboot
chmod -R 777 /var/lib/tftpboot
```

- Cambiar configuración de SELINUX para habilitar el tráfico TFTP. (RHEL o CentOS)

```
setsebool -P tftp_anon_write 1
setsebool -P tftp_home_dir 1
```

- Reiniciar servicio tftp

  - Ubuntu

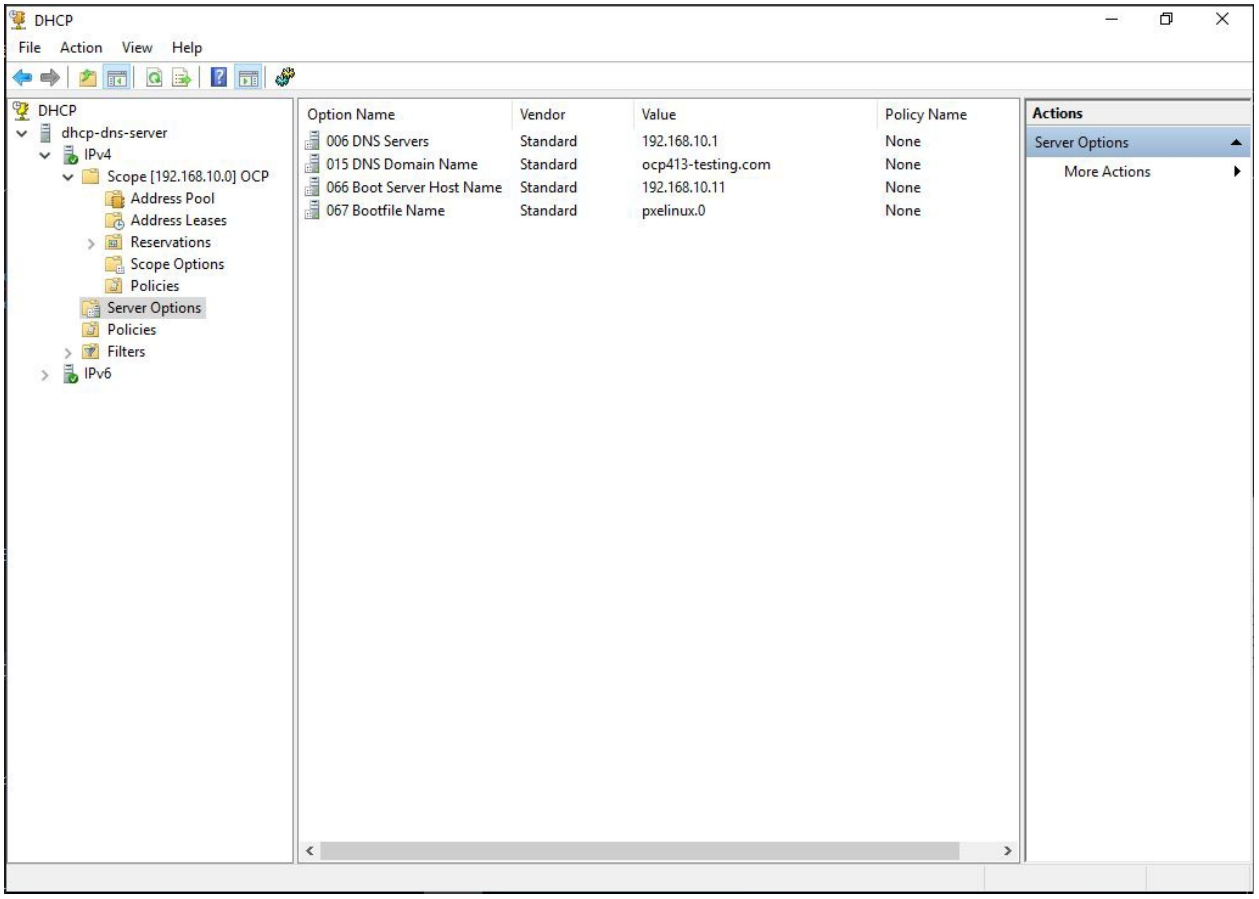
```
systemctl restart tftpd-hpa
```

  - RHEL o CentOS

```
systemctl restart tftp-server
```

- Configurar Servidor DHCP. Añadir server options 066 y 067 con la IP del tftp server (Ip de la máquina ubuntu en este caso) y el nombre del archivo de booteo, como se muestra en la imágen

  - Windows Server 2016



  - Linux

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#

default-lease-time 3600;
max-lease-time 7200;
authoritative;
allow booting;
allow bootp;

subnet 192.168.51.0 netmask 255.255.255.0 {
 option domain-search "ocp4waiops.entellab.com";
 range 192.168.51.10 192.168.51.50;
 option routers 192.168.51.1;
 option subnet-mask 255.255.255.0;
 option domain-name-servers 192.168.51.10;
}

host registry {
 hardware ethernet 00:50:56:03:00:f0;
 fixed-address 192.168.51.51;
}

host bootstrap {
 hardware ethernet 00:50:56:03:00:f4;
 fixed-address 192.168.51.100;
 next-server 192.168.51.10;
 filename "pxelinux.0";
}
```

6. Configurar servidor HTTP Apache en servidor Bastión.

- Ir al directorio de configuración de Apache.
  - Ubuntu

```
cd /etc/apache2/sites-available
```

  - RHEL o CentOS

```
cd /etc/httpd/conf
```

- Modificar el archivo de configuración default de Apache. OBS: Se puede crear un sitio nuevo creando un archivo de configuración nuevo.
  - Ubuntu

```
nano 000-default.conf

<VirtualHost 192.168.51.10:80>
    ServerAdmin root@bastion.ocp4waiops.entellab.com
    DocumentRoot /var/www/html
    ServerName bastion.ocp4waiops.entellab.com
    ErrorLog ${APACHE_LOG_DIR}/images-server-error.log
    CustomLog ${APACHE_LOG_DIR}/images-server-access.log common
    <Directory /pxeboot/boot >
        Options Indexes MultiViews
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

  - RHEL o CentOS

```
    nano httpd.conf

#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
```

```
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 192.168.51.10:80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
Include conf.modules.d/*.conf

#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User apache
Group apache

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
```

```
# virtual host being defined.
#

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
ServerAdmin root@bastion.ocp4waiops.entellab.com

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName bastion.ocp4waiops.entellab.com:80

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"

#
# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride All
    # Allow open access:
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

<Directory "/var/www/html/pxeboot/boot" >
    Options Indexes MultiViews
    AllowOverride All
    Require all granted
</Directory>


#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>

#
```

```
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
    Require all denied
</Files>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog "logs/error_log"

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    #CustomLog "logs/access_log" common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    #
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
    # If you include a trailing / on /webpath then the server will
    # require it to be present in the URL.  You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.

    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
```

```
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to the
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

</IfModule>

#
# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
    TypesConfig /etc/mime.types

    #
    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.
    #
    #AddType application/x-gzip .tgz
    #
    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.
    #
    #AddEncoding x-compress .Z
    #AddEncoding x-gzip .gz .tgz
    #
    # If the AddEncoding directives above are commented-out, then you
    # probably should define those extensions to indicate media types:
    #
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    #
    # AddHandler allows you to map certain file extensions to "handlers":
    # actions unrelated to filetype. These can be either built into the server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    #AddHandler cgi-script .cgi

    # For type maps (negotiated resources):
    #AddHandler type-map var

    #
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>

#
# Specify a default charset for all content served; this enables
# interpretation of all content as UTF-8 by default.  To use the
# default browser choice (ISO-8859-1), or to allow the META tags
# in HTML content to override this choice, comment out this
# directive:
#
AddDefaultCharset UTF-8
```

```
<IfModule mime_magic_module>
    #
    # The mod_mime_magic module allows the server to use various hints from the
    # contents of the file itself to determine its type.  The MIMEMagicFile
    # directive tells the module where the hint definitions are located.
    #
    MIMEMagicFile conf/magic
</IfModule>

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#

#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files.  This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
```

- Crear directorio */pxeboot/boot/* dentro de la ruta */var/www/html*.

```
mkdir pxeboot

cd /pxeboot

mkdir boot

cd /boot
```

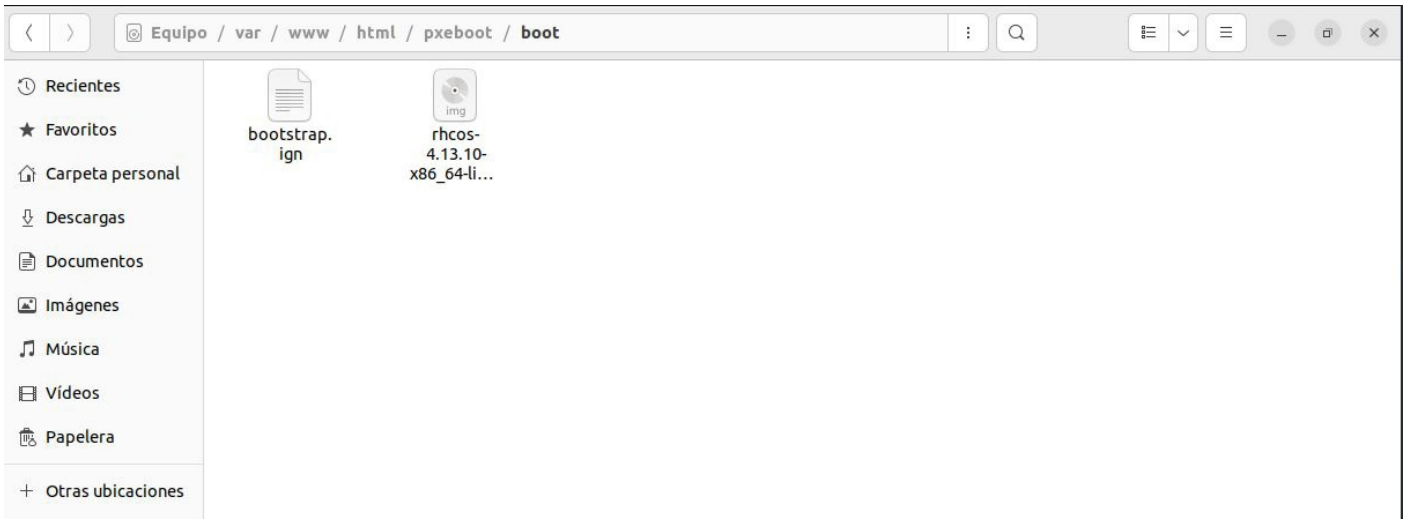
- Mover o copiar archivos "bootstrap.ign" y "rhcos-4.13.10-x86*64-live-rootfs.x86_64.img" dentro del directorio _boot*.

  El directorio debería quedar de la siguiente manera:

```
var
|___www
    |___html
        |___pxeboot
            |___boot
                |   bootstrap.ign
                |   rhcos-4.13.10-x86_64-live-rootfs.x86_64.img
```

- **OBS: Las razones por las cuales estos archivos no van en el directorio tftpboot son las siguientes:**
  - El instalador del SO RCHOS solicita sí o sí que el archivo "rhcos-4.13.10-x86_64-live-rootfs.x86_64.img" sea transferido a través de HTTP,TFTP o HTTPS.
  - La solución más rapida es HTTP ya que la TFTP demoró alrededor de 2 horas en obtener el archivo. (El archivo pesa 1 GB aprox.).
  - Existen más opciones de seguridad al realizar la transferencia por HTTP o HTTPS que no fueron exploradas en éste ejercicio.
  - Al momento de iniciar el booteo del SO RCHOS, la máquina, por alguna razón que desconozco, bloquea las llamadas entrantes desde el servidor donde están alojadas las imágenes y solo se pueden obtener directamente desde el directorio /tftpboot. Los 2 archivos escenciales para que comienze la instalación del SO RHCOS, son "rhcos-4.13.10-x86_64-live-kernel-x86_64" y "rhcos-

4.13.10-x86_64-custom-initramfs.x86_64.img".



- Quitarle permisos al directorio */var/www/html/pxeboot/boot* y contenido.

  OBS: Para efecto de este ejercicio se le quitaron todos los permisos las rutas, no es la mejor opción pero aun no está claro qué permisos necesita el servidor HTTP para servir los archivos a la máquina booteable.

  -Ubuntu

```
chown -R nobody:nogroup /var/www

chmod -R 777 /var/www
```

  - RHEL o CentOS

```
chown -R nobody:nobody /var/www

chmod -R 777 /var/www
```

- Configurar firewall para el servidor TFTP y HTTP (Apache).

  - RHEL o CentOS

```
yum install -y ufw
```

  OBS: Estos puertos son los predeterminados para los protocolos TFTP y HTTP. En teoría, si es requerido, se puede configurar cada uno para que sirva en un puerto distinto.

  - Ubuntu

```
ufw allow Apache

ufw allow from any to any proto udp port 69

ufw reload

ufw status
```

  - RHEL o CentOS

```
ufw allow http

ufw allow tftp

ufw reload

ufw status
```

- Configurar SELINUX para permitir el tráfico HTTP. (RHEL o CentOS)

- Verificar si están los puertos correctamente habilitados

```
semanage port -l | grep http
```

- Restaurar los contextos de seguridad de SELinux

```
restorecon -Rv /var/
```

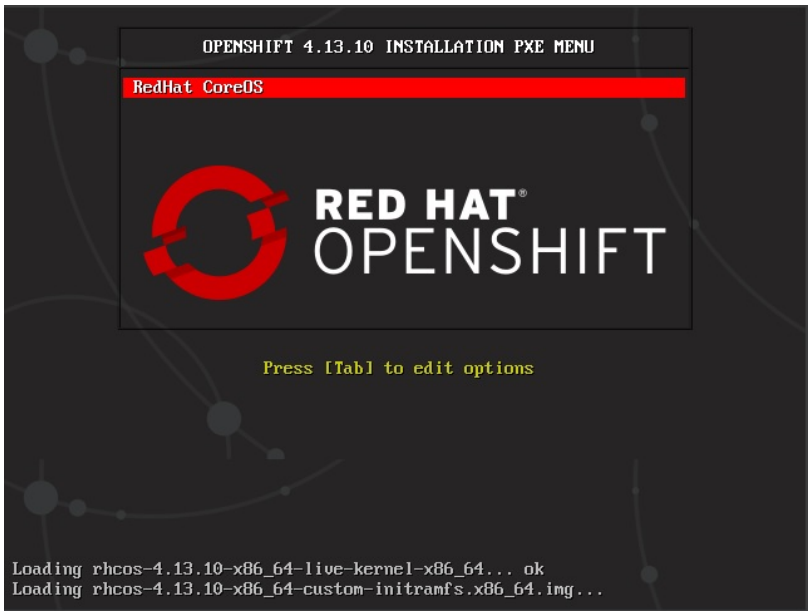
```
root@bastion:~# ufw status
Estado: activo

Hasta                      Acción      Desde
-----                      ------      -----
Anywhere on enp0s8         DENY        192.168.10.0/24
22                         ALLOW       Anywhere
8080                       ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
69/udp                     ALLOW       Anywhere
Apache                     ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
8080 (v6)                  ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
69/udp (v6)                ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

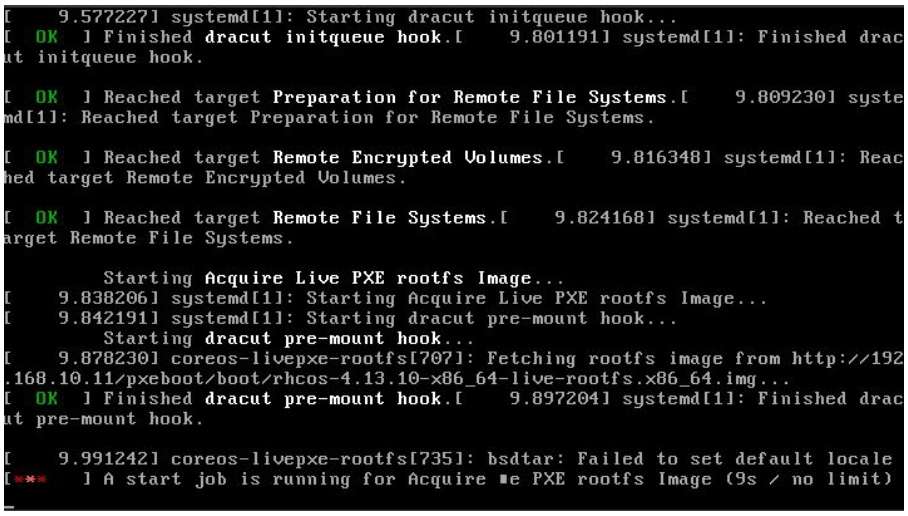
7. Configurar archivo default en el directorio */var/lib/tftpboot/pxelinux.cfg*.

```
UI vesamenu.c32
MENU BACKGROUND        bg-ocp.png
MENU COLOR sel         4  #ffffff std
MENU COLOR title       1  #ffffff
MENU TITLE OPENSHIFT 4.13.10 INSTALLATION PXE MENU
LABEL INSTALL BOOTSTRAP RedHat CoreOS
        MENU LABEL INSTALL BOOTSTRAP RedHat CoreOS
        KERNEL rhcos-4.13.10-x86_64-live-kernel-x86_64
        APPEND initrd=rhcos-4.13.10-x86_64-custom-initramfs.x86_64.img
coreos.live.rootfs_url=http://192.168.51.10/pxeboot/boot/rhcos-4.13.10-x86_64-live-rootfs.x86_64.img ignition.firstboot
ignition.platform.id=metal
LABEL INSTALL WORKER RedHat CoreOS
        MENU LABEL INSTALL WORKER RedHat CoreOS
        KERNEL rhcos-4.13.10-x86_64-live-kernel-x86_64
        APPEND initrd=rhcos-4.13.10-x86_64-custom-worker-initramfs.x86_64.img
coreos.live.rootfs_url=http://192.168.51.10/pxeboot/boot/rhcos-4.13.10-x86_64-live-rootfs.x86_64.img ignition.firstboot
ignition.platform.id=metal
LABEL INSTALL MASTER RedHat CoreOS
        MENU LABEL INSTALL MASTER RedHat CoreOS
        KERNEL rhcos-4.13.10-x86_64-live-kernel-x86_64
        APPEND initrd=rhcos-4.13.10-x86_64-custom-master-initramfs.x86_64.img
coreos.live.rootfs_url=http://192.168.51.10/pxeboot/boot/rhcos-4.13.10-x86_64-live-rootfs.x86_64.img ignition.firstboot
ignition.platform.id=metal
```

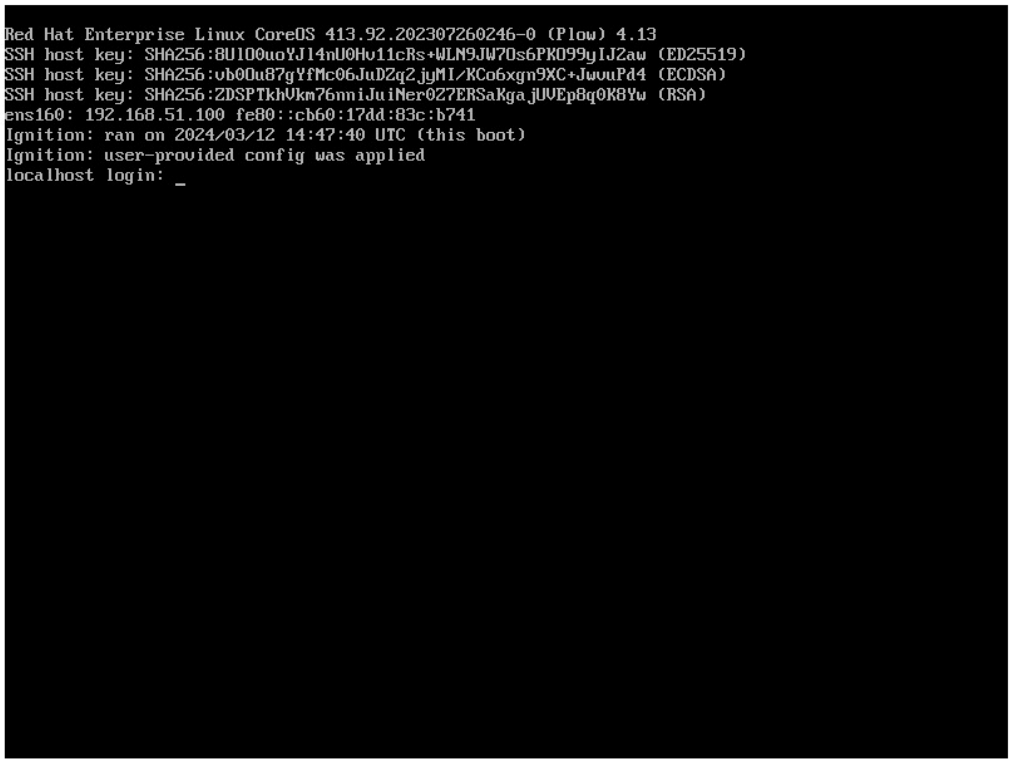
8. Iniciar la máquina virtual. Al bootear debería aparecer el siguiente menú.

```
iPXE (PCI E2:00.0) starting execution...ok
iPXE initialising devices...ok



iPXE 1.21.1 -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS TFTP PXE PXEXT

net0: 08:00:27:1b:dc:48 using 82540em on 0000:00:03.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086101)]
Waiting for link-up on net0..... ok
Configuring (net0 08:00:27:1b:dc:48)...... ok
net0: 192.168.10.13/255.255.255.0
Next server: 192.168.10.11
Filename: pxelinux.0
tftp://192.168.10.11/pxelinux.0... ok
pxelinux.0 : 42584 bytes [PXE-NBP]

PXELINUX 6.04 PXE 20210811 Copyright (C) 1994-2015 H. Peter Anvin et al
_
```





Para ingresar al nodo, se debe ingresar por SSH a la máquina para cambiar la contraseña del usuario "core"



- Conexión al nodo bootstrap por ssh desde el Bastion con la IP. (Esto es posible gracias al definir la clave rsa publica de la máquina bastion en el archivo "install-config.yaml" a la hora de crear los ignition files).

```
[root@bastion httpd]# ssh core@192.168.51.100
The authenticity of host '192.168.51.100 (192.168.51.100)' can't be established.
ECDSA key fingerprint is SHA256:vb0Ou87gYfMc06JuDZq2jyMI/KCo6xgn9XC+JwvuPd4.
ECDSA key fingerprint is MD5:4d:65:56:0f:cb:d1:f2:98:33:4b:de:19:6f:2d:0f:20.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.51.100' (ECDSA) to the list of known hosts.
Red Hat Enterprise Linux CoreOS 413.92.202307260246-0
  Part of OpenShift 4.13, RHCOS is a Kubernetes native operating system
  managed by the Machine Config Operator (`clusteroperator/machine-config`).

WARNING: Direct SSH access to machines is not recommended; instead,
make configuration changes via `machineconfig` objects:
  https://docs.openshift.com/container-platform/4.13/architecture/architecture-rhcos.html

---
This is the bootstrap node; it will be destroyed when the master is fully up.

The primary services are release-image.service followed by bootkube.service. To watch their status, run e.g.

  journalctl -b -f -u release-image.service -u bootkube.service
Last failed login: Tue Mar 12 14:51:22 UTC 2024 on tty1
There was 1 failed login attempt since the last successful login.
[core@localhost ~]$
```

- Conexión al nodo bootstrap por ssh desde el Bastion con el nombre de dominio.

```
[root@bastion httpd]# ssh core@bootstrap.ocp4waiops.entellab.com
The authenticity of host 'bootstrap.ocp4waiops.entellab.com (192.168.51.100)' can't be established.
ECDSA key fingerprint is SHA256:vb0Ou87gYfMc06JuDZq2jyMI/KCo6xgn9XC+JwvuPd4.
ECDSA key fingerprint is MD5:4d:65:56:0f:cb:d1:f2:98:33:4b:de:19:6f:2d:0f:20.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bootstrap.ocp4waiops.entellab.com' (ECDSA) to the list of known hosts.
Red Hat Enterprise Linux CoreOS 413.92.202307260246-0
  Part of OpenShift 4.13, RHCOS is a Kubernetes native operating system
  managed by the Machine Config Operator (`clusteroperator/machine-config`).

WARNING: Direct SSH access to machines is not recommended; instead,
make configuration changes via `machineconfig` objects:
  https://docs.openshift.com/container-platform/4.13/architecture/architecture-rhcos.html

---
This is the bootstrap node; it will be destroyed when the master is fully up.

The primary services are release-image.service followed by bootkube.service. To watch their status, run e.g.

  journalctl -b -f -u release-image.service -u bootkube.service
Last login: Tue Mar 12 14:58:22 2024 from 192.168.51.10
[core@localhost ~]$
```

- Evidencias de resolución del DNS para la máquina bootstrap

```
[root@bastion httpd]# dig bootstrap.ocp4waiops.entellab.com

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el7 <<>> bootstrap.ocp4waiops.entellab.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38144
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;bootstrap.ocp4waiops.entellab.com. IN  A

;; ANSWER SECTION:
bootstrap.ocp4waiops.entellab.com. 10800 IN A   192.168.51.100

;; AUTHORITY SECTION:
entellab.com.           10800   IN      NS      dns.entellab.com.

;; ADDITIONAL SECTION:
dns.entellab.com.       10800   IN      A       192.168.51.10

;; Query time: 0 msec
;; SERVER: 192.168.51.10#53(192.168.51.10)
;; WHEN: Tue Mar 12 14:55:52 UTC 2024
;; MSG SIZE  rcvd: 112
```

```
[root@bastion httpd]# nslookup 192.168.51.100
100.51.168.192.in-addr.arpa     name = bootstrap.ocp4waiops.entellab.com.

[root@bastion httpd]# nslookup bootstrap.ocp4waiops.entellab.com
Server:         192.168.51.10
Address:        192.168.51.10#53

Name:   bootstrap.ocp4waiops.entellab.com
Address: 192.168.51.100
```

NODO BOOTSTRAP OBS

- En este ejercicio, solo se tenía las máquinas (DHCP,BASTION y BOOTSTRAP) levantadas, por lo que al bootear el nodo Bootstrap, éste intento hacer un pull de las imágenes al registry que se configuró previamente, por lo que en producción, cuando se instalen y configuren los nodos, todas las máquinas deben estar operativas.

4. HAProxy Load Balancer

- En este ejercicio se utilizó una máquina virtual CentOS 7 para el Load Balancer.

1. Levanta máquina con conexión a Internet para instalar HAProxy.

```
yum install haproxy
```

```
systemctl start haproxy

systemctl enable haproxy

systemctl status haproxy
```

2. Una vez instalado el paquete, se puede dar de baja la interfaz que da acceso a internet.

3. Configurar el archivo de configuración de HAProxy.

```
nano /etc/haproxy/haproxy.cfg
```

4. Para éste ejercicio la configuración sería la siguiente:

```
global
    log         192.168.51.10 local2
    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    stats socket /run/haproxy/admin.sock mode 600 level admin
    stats socket /var/lib/haproxy/stats
    daemon
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000
listen stats
    bind :9000
    mode http
    stats enable
    stats uri /
frontend ocp4waiops-entellab-k8s-api-fe
    bind :6443
    default_backend ocp4waiops-entellab-k8s-api-be
    mode tcp
    option tcplog
backend ocp4waiops-entellab-k8s-api-be
    bind *:6443
    mode tcp
    option  httpchk GET /readyz HTTP/1.0
    option  log-health-checks
    balance roundrobin
```

```
    server bootstrap bootstrap.ocp4waiops.entellab.com:6443 verify none check check-ssl inter 10s fall 2 rise 3 backup
    server master1 master1.ocp4waiops.entellab.com:6443 weight 1 verify none check check-ssl inter 10s fall 2 rise 3
    server master2 master2.ocp4waiops.entellab.com:6443 weight 1 verify none check check-ssl inter 10s fall 2 rise 3
    server master3 master3.ocp4waiops.entellab.com:6443 weight 1 verify none check check-ssl inter 10s fall 2 rise 3
frontend ocp4waiops-entellab-machine-config-server-fe
    bind :22623
    default_backend ocp4waiops-entellab-machine-config-server-be
    mode tcp
    option tcplog
backend ocp4waiops-entellab-machine-config-server-be
    bind *:22623
    mode tcp
    server bootstrap bootstrap.ocp4waiops.entellab.com:22623 check inter 1s backup
    server master1 master1.ocp4waiops.entellab.com:22623 check inter 1s
    server master2 master2.ocp4waiops.entellab.com:22623 check inter 1s
    server master3 master3.ocp4waiops.entellab.com:22623 check inter 1s
frontend ocp4waiops-entellab-https-ingress-traffic-fe
    bind :443
    default_backend ocp4waiops-entellab-ingress-router-https-be
    mode tcp
    option tcplog
backend ocp4waiops-entellab-ingress-router-https-be
    bind *:443
    mode tcp
    balance source
    server worker1 worker1.ocp4waiops.entellab.com:443 check inter 1s
    server worker2 worker2.ocp4waiops.entellab.com:443 check inter 1s
frontend ocp4waiops-entellab-http-ingress-traffic-fe
    bind :80
    default_backend ocp4waiops-entellab-ingress-router-http-be
    mode tcp
    option tcplog
backend ocp4waiops-entellab-ingress-router-http-be
    bind *:80
    mode tcp
    balance source
    server worker1 worker1.ocp4waiops.entellab.com:80 check inter 1s
    server worker2 worker2.ocp4waiops.entellab.com:80 check inter 1s
```

- Reiniciar servicio haproxy

```
    systemctl restart haproxy
```