



Cisco Prime Network Registrar on Container

Cisco Prime Network Registrar 11.0 can be run as a Docker container that you can install in your own infrastructure.

The following Docker images are provided for Cisco Prime Network Registrar 11.0:

- Regional container: `cpnr-regional-11.0-1.el8.x86_64_rhel_docker.tar.gz`
- Local container: `cpnr-local-11.0-1.el8.x86_64_rhel_docker.tar.gz`



Note The name of the images will change with releases in future.

- [Requirements on the Host Machine, on page 1](#)
- [Running Cisco Prime Network Registrar Docker Container, on page 1](#)
- [Moving an Existing Cisco Prime Network Registrar Cluster to Docker Container, on page 4](#)

Requirements on the Host Machine

- Identify the ports on the host machine to be exposed to ports required by Cisco Prime Network Registrar container. For a complete list of ports used by Cisco Prime Network Registrar services, see the *"Default Ports for Cisco Prime Network Registrar Services"* section in the *Cisco Prime Network Registrar 11.0 Administration Guide*.
- Select an option to persist data of Cisco Prime Network Registrar container on the host machines: Bind mount (where a directory on the host machine is used) or Volume (which is managed by Docker).
- For IPv4, you can use either bridged or macvlan network; Cisco recommends macvlan for better performance.
- For IPv6, you will need to configure the container to have an IPv6 address.

Running Cisco Prime Network Registrar Docker Container

To run Cisco Prime Network Registrar as Docker container, you must first download the Docker image of your choice. Then, do the following:

Step 1 Load the Docker image using the following command:

- For the regional container:

```
# docker load -i cpmr-regional-11.0-1.e18.x86_64_rhel_docker.tar.gz
```

- For the local container:

```
# docker load -i cpmr-local-11.0-1.e18.x86_64_rhel_docker.tar.gz
```

Step 2 Verify that the image is successfully loaded using the following command:

```
# docker image ls
```

Step 3 Run the Docker container using the following command:

- For the regional container:

```
# docker run -d --name cpmr_regional_container --privileged=true -p 8453:8453 -p 1244:1244 --mount
type=bind,source=/data/cpmr_regional_data,target=/var/nwreg2/regional cpmr-regional:11.0
/usr/sbin/init
```

In the above command:

- Default bridge networking driver of Docker is used. Ports needed by container are exposed—8453 is for regional web UI, 1244 for regional configuration management.
- Data directory of Cisco Prime Network Registrar is /var/nwreg2/regional and mountpoint on the host is /data/cpmr_regional_data
- The command to be run is /usr/sbin/init

If it is required to synchronize the timezone of the host and the Docker container, then add the **-v /etc/localtime:/etc/localtime** option to the above Docker run command.

By default, the core files are available in the /var/lib/systemd/coredump directory of the Docker host machine. To collect the core files via the **cpmr_tactool** utility, run the following commands on the Docker host machine:

```
# echo '/data/cpmr_regional_data/core.%p' > /proc/sys/kernel/core_pattern'
# ulimit -c unlimited
```

After running the above commands, the core files will be available in the /data/cpmr_regional_data directory and you can use **cpmr_tactool** to collect them.

- For the local container:

```
# docker run -d --name cpmr_local_container --privileged=true -p 8443:8443 -p 1234:1234 -p
67:67/udp -p 53:53/udp --mount type=bind,source=/data/cpmr_local1_data,target=/var/nwreg2/local
cpmr-local:11.0 /usr/sbin/init
```

In the above command:

- Default bridge networking driver of Docker is used. Ports needed by container are exposed—8443 is for the web UI, 1234 for local configuration management, 67 for DHCP, and 53 for DNS. For other services like SNMP, TFTP see the *"Default Ports for Cisco Prime Network Registrar Services"* section in the *Cisco Prime Network Registrar 11.0 Administration Guide*.
- Data directory of Cisco Prime Network Registrar is /var/nwreg2/local and mountpoint on the host is /data/cpmr_local1_data
- The command to be run is /usr/sbin/init

If it is required to synchronize the timezone of the host and the Docker container, then add the **-v /etc/localtime:/etc/localtime** option to the above Docker run command.

By default, the core files are available in the `/var/lib/systemd/coredump` directory of the Docker host machine. To collect the core files via the **cnr_tactool** utility, run the following commands on the Docker host machine:

```
# echo '/data/cpnr_local1_data/core.%p' > /proc/sys/kernel/core_pattern'
# ulimit -c unlimited
```

After running the above commands, the core files will be available in the `/data/cpnr_local1_data` directory and you can use **cnr_tactool** to collect them.

Step 4 Start configuring Cisco Prime Network Registrar.

- For the regional container:

- To connect using web UI, use `https://hostip:8453`
- To connect using CLI, use the following command:

```
install-path/usrbin/nrcmd -R -C hostip:1244 -N username -P password
```

- For the local container:

- To connect using web UI, use `https://hostip:8443`
- To connect using CLI, use the following command:

```
install-path/usrbin/nrcmd -C hostip:1234 -N username -P password
```

For running DHCP failover and HA DNS, we recommend running two Cisco Prime Network Registrar containers (Main and Backup) in separate hosts, as this avoids single point of failure. Given that bridge network is limited to a single host, using macvlan as networking driver is the best choice. With macvlan, container appears to be directly connected to physical network.

If the Docker daemon allows IPv6, you can use dual-stack macvlan networks, that is, both IPv4 and IPv6:

```
# docker network create --driver=macvlan --ipv6 --subnet=2001:db8:1:1::/64
--gateway=2001:db8:1:1::1 --subnet=10.0.0.0/24 --gateway=10.0.0.1 -o macvlan_mode=bridge
-o parent=eth0 cpnr_macvlan
```

Run Cisco Prime Network Registrar container and attach it to the macvlan network created above:

```
# docker run -d --name cpnr_dhcp_main --network=cpnr_macvlan --ip 10.0.0.20 --ip6
2001:db8:1:1::20 --privileged=true --mount type=bind,source=/data/cpnr_dhcp_main_data,
target=/var/nwreg2/local cpnr-local:11.0 /usr/sbin/init
```

This Cisco Prime Network Registrar container (local) can now be reached at `10.0.0.20` and `2001:db8:1:1::20`.

- To connect using web UI over IPv4, use `https://10.0.0.20:8443`
- To connect using CLI over IPv6, use the following command:

```
install-path/usrbin/nrcmd -C [2001:db8:1:1::20]:1234 -N username -P password
```

Moving an Existing Cisco Prime Network Registrar Cluster to Docker Container

To move to Cisco Prime Network Registrar 11.0 Docker container from an existing Cisco Prime Network Registrar 8.3 or later cluster, do the following:

Step 1 Remove the existing installation using the procedure described in [Uninstalling Cisco Prime Network Registrar](#).

Step 2 Delete the /opt/nwreg2 folder. Ensure NOT to delete the /var/nwreg2 folder after uninstallation.

If you are upgrading to Cisco Prime Network Registrar 11.0 Docker Container on the same machine, then skip this Step 3 and proceed to Step 4.

Step 3 If you are upgrading to Cisco Prime Network Registrar 11.0 Docker Container on a different machine, then create the source directory tree (for example, /data/cpnr_local1_data for local cluster and /data/cpnr_regional_data for regional cluster) in the machine where you want to create the Docker instance (target machine). Then, transfer the original cluster's /var/nwreg2/{local | regional} directory to this directory. Use the following commands:

- For regional cluster:

```
# mkdir -p /data/cpnr_regional_data
# mv /var/nwreg2/regional /data/cpnr_regional_data
```

- For local cluster:

```
# mkdir -p /data/cpnr_local1_data
# mv /var/nwreg2/local /data/cpnr_local1_data
```

Note Copy the cnr.conf file from the /opt/nwreg2/{local | regional}/conf directory to the conf folder in the source directory of the target machine. Use the following commands:

- For regional cluster:

```
# mv /opt/nwreg2/regional/conf /data/cpnr_regional_data/conf
```

- For local cluster:

```
# mv /opt/nwreg2/local/conf /data/cpnr_local1_data/conf
```

Step 4 Create the Docker instance using the following command:

- For regional container:

```
$ docker run -d --name cpnr_container -v /etc/localtime:/etc/localtime --network=mymacvlan
--ip hostip --ip6 ipv6address --privileged=true --hostname=hostip --mount type=bind,
source=/data/cpnr_regional_data,target=/var/nwreg2/regional cpnr_regional:11.0 /usr/sbin/init
```

- For local container:

```
$ docker run -d --name cpnr_container -v /etc/localtime:/etc/localtime --network=mymacvlan
--ip hostip --ip6 ipv6address --privileged=true --hostname=hostip --mount type=bind,
source=/data/cpnr_local1_data,target=/var/nwreg2/local cpnr_local:11.0 /usr/sbin/init
```

Step 5 Verify if the previous configuration, including scopes and zones, is intact in Cisco Prime Network Registrar 11.0 server. Also, verify that the data.bak folder is created which contains the pre-upgrade version's database version as backup.



Note After performing the above steps, all settings will take their defaults and you may have to take additional steps to reinstall the certificates or change ports. For information, see [Installing Your Own Certificate for Web UI Access](#).
