



コンテナでの Cisco Prime Network Registrar

Cisco Prime Network Registrar 11.0 は、独自のインフラストラクチャにインストールできる Docker コンテナとして実行できます。

Cisco Prime Network Registrar 11.0 では、次の Docker イメージが提供されます。

- リージョンコンテナ : `cpnr-regional-11.0-1.el8.x86_64_rhel_docker.tar.gz`
- ローカルコンテナ : `cpnr-local-11.0-1.el8.x86_64_rhel_docker.tar.gz`



(注) イメージの名前は、今後のリリースで変更されます。

- [ホストマシンの要件 \(1 ページ\)](#)
- [Cisco Prime Network Registrar Docker コンテナの実行 \(2 ページ\)](#)

ホストマシンの要件

- Cisco Prime Network Registrar コンテナが必要とするポートに公開するホストマシン上のポートを特定します。Cisco Prime Network Registrar サービスで使用されるポートの完全なリストについては、*Cisco Prime Network Registrar 11.0* アドミニストレーションガイドの「*Cisco Prime Network Registrar* サービスのデフォルトポート」の項を参照してください。
- ホストマシン上の Cisco Prime Network Registrar コンテナのデータを保持するオプションを [バインドマウント (Bind mount)] (ホストマシン上のディレクトリが使用されます) または [ボリューム (Volume)] (Docker によって管理されます) のいずれかから選択します。
- IPv4 の場合は、ブリッジネットワークまたは macvlan ネットワークを使用できます。パフォーマンス向上のため、macvlan を推奨します。
- IPv6 の場合は、IPv6 アドレスを持つようにコンテナを設定する必要があります。

Cisco Prime Network Registrar Docker コンテナの実行

Cisco Prime Network Registrar を Docker コンテナとして実行するには、最初に選択した Docker イメージをダウンロードする必要があります。次に、以下の手順を実行します。

ステップ 1 次のコマンドを使用して、Docker イメージを読み込みます。

- リージョナルコンテナの場合：

```
# docker load -i cpnr-regional-11.0-1.el8.x86_64_rhel_docker.tar.gz
```

- ローカルコンテナの場合：

```
# docker load -i cpnr-local-11.0-1.el8.x86_64_rhel_docker.tar.gz
```

ステップ 2 次のコマンドを使用して、イメージが正常に読み込まれていることを確認します。

```
# docker image ls
```

ステップ 3 次のコマンドを使用して Docker コンテナを実行します。

- リージョナルコンテナの場合：

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

上記のコマンドでは、次のようになります。

- Docker のデフォルトブリッジネットワークングドライバが使用されます。コンテナに必要なポートが公開されます。8453 はリージョナルの Web UI 用で、1244 はリージョナルの設定管理用です。
- Cisco Prime Network Registrar のデータディレクトリは `var/nwreg2/regional` で、ホストのマウントポイントは `/data/cpnr_regional_data` です。
- 実行するコマンドは `/usr/sbin/init` です。

ホストと Docker コンテナのタイムゾーンを同期する必要がある場合は、上記の Docker run コマンドに `-v /etc/localtime:/etc/localtime` オプションを追加します。

デフォルトでは、コアファイルは Docker ホストマシンの `/var/lib/systemd/coredump` ディレクトリにあります。`cnr_tactool` ユーティリティを使用してコアファイルを収集するには、Docker ホストマシンで次のコマンドを実行します。

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

上記のコマンドを実行すると、コアファイルが `/data/cpnr_regional_data` ディレクトリで使用可能になり、`cnr_tactool` を使用して収集できるようになります。

- ローカルコンテナの場合：

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

上記のコマンドでは、次のようになります。

- Docker のデフォルトブリッジネットワークングドライバが使用されます。コンテナに必要なポートが公開されます。8443 は Web UI 用、1234 はローカルの設定管理用、67 は DHCP 用、53 は DNS 用です。SNMP や TFTP などの他のサービスについては、*Cisco Prime Network Registrar 11.0* アドミニストレーションガイドの「*Cisco Prime Network Registrar* サービスのデフォルトポート」の項を参照してください。
- Cisco Prime Network Registrar のデータディレクトリは `/var/nwreg2/local` で、ホストのマウントポイントは `/data/cpnr_local1_data` です。
- 実行するコマンドは `/usr/sbin/init` です。

ホストと Docker コンテナのタイムゾーンを同期する必要がある場合は、上記の Docker run コマンドに `-v /etc/localtime:/etc/localtime` オプションを追加します。

デフォルトでは、コアファイルは Docker ホストマシンの `/var/lib/systemd/coredump` ディレクトリにあります。`cnr_tactool` コーティリティを使用してコアファイルを収集するには、Docker ホストマシンで次のコマンドを実行します。

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

上記のコマンドを実行すると、コアファイルが `/data/cpnr_local1_data` ディレクトリで使用可能になり、`cnr_tactool` を使用して収集できるようになります。

ステップ 4 Cisco Prime Network Registrar の設定を開始します。

- リージョナルコンテナの場合：
 - Web UI を使用して接続するには、`https://hostip:8453` を使用します。
 - CLI を使用して接続するには、次のコマンドを使用します。


```
install-path/usrbin/nrcmd -R -C hostip:1244 -N username -P password
```
- ローカルコンテナの場合：
 - Web UI を使用して接続するには、`https://hostip:8443` を使用します。
 - CLI を使用して接続するには、次のコマンドを使用します。


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

DHCP フェールオーバーと HA DNS を実行する場合は、2 つ Cisco Prime Network Registrar のコンテナ（メインとバックアップ）を別々のホストで実行することをお勧めします。これにより、シングルポイント障害を回避できます。ブリッジネットワークが単一のホストに制限されている場合は、ネットワークドライバとして `macvlan` を使用するのが最適な選択です。`macvlan` では、コンテナは物理ネットワークに直接接続されているように見えます。

Docker デーモンで IPv6 が許可されている場合は、デュアルスタック macvlan ネットワーク、つまり IPv4 と IPv6 の両方を使用できます。

```
# 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
```

Cisco Prime Network Registrar コンテナを実行し、上記で作成した macvlan ネットワークに接続します。

```
# 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
```

この Cisco Prime Network Registrar コンテナ（ローカル）は、10.0.0.20 および 2001:db8:1:1::20 で到達可能です。

- IPv4 経由の Web UI を使用して接続するには、<https://10.0.0.20:8443> を使用します。
- CLI over IPv6 を使用して接続するには、次のコマンドを使用します。

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