

# NX-OS Bash 셸 DNS 컨피그레이션

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[Bash 셸 DNS 컨피그레이션](#)

[Bash 셸 DNS 확인](#)

[1단계. 테스트할 유효한 네트워크 네임스페이스의 사용을 확인합니다.](#)

[2단계. 테스트 호스트의 호스트 이름을 사용하여 DNS 확인 기능을 확인합니다.](#)

[/etc/resolv.conf 파일 형식](#)

[예](#)

[관련 정보](#)

## 소개

이 문서에서는 DNS 호스트 이름을 IP 주소로 확인할 수 있도록 Bash 내에서 DNS 서버를 구성하는데 사용되는 단계에 대해 설명합니다.

Cisco Nexus 3000 및 9000 Series 장치는 Bash(Bourne-Again SHell)를 통해 NX-OS의 기본 Linux 시스템에 액세스할 수 있습니다. Bash는 Linux 환경을 통해 시스템 관리 및 모니터링을 활성화합니다. NX-OS의 Bash에 대한 자세한 내용은 [Cisco Nexus 9000 Series NX-OS Programmability Guide의 Bash 장을 참조하십시오.](#)

Bash 셸에서 일반 작업을 수행하는 동안 사용자에게 친숙한 도메인 이름을 숫자 IP 주소로 변환해야 할 수도 있습니다. 이러한 작업에는 curl 또는 wget 유틸리티를 웹 서버의 리소스에 액세스하거나 docker pull 명령을 사용하여 Docker 이미지를 다운로드하는 작업이 포함됩니다.

## 사전 요구 사항

### 요구 사항

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

**참고:** Bash 셸은 Cisco Nexus 디바이스에서 활성화됩니다. Bash 셸을 활성화하는 지침은 [Cisco Nexus 9000 Series NX-OS Programmability Guide](#)의 Bash 장의 "Accessing Bash" 섹션을 참조하십시오.

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- NX-OS 릴리스 6.1(2)I2(1)부터 시작하는 Nexus 9000 플랫폼

- NX-OS 릴리스 6.0(2)U4(1)부터 시작하는 Nexus 3000 플랫폼

이 문서의 정보는 특정 랩 환경의 디바이스에서 생성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## Bash 셸 DNS 컨피그레이션

Bash 셸을 통해 액세스하는 Linux 환경은 `/etc/resolv.conf` 파일을 사용하여 대부분의 다른 Unix와 유사한 운영 체제와 마찬가지로 DNS 구성을 저장합니다.

1. `run bash sudo su -` 명령을 통해 루트 사용자로 Bash 셸에 로그인합니다.

```
Nexus# run bash sudo su -
root@Nexus#whoami
root
```

2. `/etc/resolv.conf` 파일의 현재 내용을 확인합니다. 이 예에서는 파일이 비어 있습니다.

```
root@Nexus#cat /etc/resolv.conf
root@Nexus#
```

3. vi 텍스트 편집기를 사용하여 편집할 `/etc/resolv.conf` 파일을 엽니다.

```
root@Nexus#vi /etc/resolv.conf
```

4. `i` 키를 눌러 INSERT 모드를 시작한 다음 원하는 구성을 입력합니다. `/etc/resolv.conf` 파일 내의 구성 형식에 대한 자세한 내용은 이 문서의 `/etc/resolv.conf` 파일 형식 섹션을 참조하십시오.

5. 파일을 수정한 후 `Esc` 키를 눌러 INSERT 모드를 종료한 다음 `:x` 입력하여 파일의 모든 변경 내용을 저장하고 닫습니다.

## Bash 셸 DNS 확인

Bash 셸의 DNS 컨피그레이션이 변경되면 도메인 이름 확인이 성공했는지 확인합니다. 도메인 이름 확인을 테스트하는 가장 간단한 방법은 도메인 호스트 이름을 대상으로 `ping` 유틸리티를 사용하는 것입니다. 이 문서에서는 `test.cisco.com`의 테스트 호스트와 192.168.2.1 및 192.168.2.2의 DNS 서버를 사용하여 유효한 DNS 컨피그레이션을 확인하는 방법을 보여 줍니다.

### 1단계. 테스트할 유효한 네트워크 네임스페이스의 사용을 확인합니다.

기본적으로 Bash 셸은 달리 지시하지 않는 한 기본 네트워크 네임스페이스를 사용합니다. 네트워크 네임스페이스는 NX-OS VRF와 논리적으로 동일하며, `ip netns` 명령은 아래와 같이 Bash 셸에서 사용할 수 있는 네임스페이스 목록을 표시합니다.

```
root@Nexus#ip netns
EXAMPLE-VRF (id: 2)
management (id: 1)
```

default (id: 0)

테스트할 수 있는 유효한 네트워크 네임스페이스는 /etc/resolv.conf 파일에 구성된 DNS 네임서버에 대한 IP 연결 및 테스트 호스트가 확인하는 IP 주소에 대한 IP 연결이 있는 네임스페이스입니다.

`ip netns exec {namespace} {desired-command}` 명령 사용하여 네임스페이스 `{namespace}`에서 명령 `{desired-command}`을 실행할 수 있습니다.또는 `ip netns exec {namespace} bash` 명령을 사용하여 특정 네임스페이스의 컨텍스트 내에서 Bash 셸을 실행할 수 있습니다.이 예에서 이전 방법론은 사용되었으며, 관리 네임스페이스가 `test.cisco.com` 호스트(192.168.2.100)과 두 DNS 서버(192.168.2.1 및 192.168.2.2)이 소유한 IP 주소와 IP 연결이 있는지 확인합니다.

```
root@Nexus#ip netns exec management ping 192.168.2.100 -c 5
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_seq=1 ttl=59 time=0.277 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=59 time=0.284 ms
64 bytes from 192.168.2.100: icmp_seq=3 ttl=59 time=0.280 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=59 time=0.274 ms
64 bytes from 192.168.2.100: icmp_seq=5 ttl=59 time=0.297 ms

--- 192.168.2.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 400lms
rtt min/avg/max/mdev = 0.274/0.282/0.297/0.017 ms
```

```
root@Nexus#ip netns exec management ping 192.168.2.1 -c 5
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=59 time=0.277 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=59 time=0.284 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=59 time=0.280 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=59 time=0.274 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=59 time=0.297 ms

--- 192.168.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 400lms
rtt min/avg/max/mdev = 0.274/0.282/0.297/0.017 ms
```

```
root@Nexus#ip netns exec management ping 192.168.2.2 -c 5
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=59 time=0.277 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=59 time=0.284 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=59 time=0.280 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=59 time=0.274 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=59 time=0.297 ms

--- 192.168.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 400lms
rtt min/avg/max/mdev = 0.274/0.282/0.297/0.017 ms
```

## 2단계. 테스트 호스트의 호스트 이름을 사용하여 DNS 확인 기능을 확인합니다.

테스트 호스트 이름의 대상과 함께 ping 유틸리티를 사용합니다.테스트 호스트에서 ICMP Replies를 수신하고 ICMP Replies에 포함된 IP 주소가 호스트 이름이 확인될 것으로 예상되는 IP 주소인 경우 DNS 확인이 Bash 셸 내에서 작동하는 것으로 확인됩니다.

이 예에서는 관리 네임스페이스 내에서 ping 유틸리티를 사용하여 올바른 DNS 확인을 확인하는 방법을 보여 줍니다.test.cisco.com의 도메인 호스트 이름이 192.168.2.100으로 확인되는 방법에 유의하십시오. 이 주소는 호스트 이름이 확인될 것으로 예상되는 IP 주소입니다.

```
root@Nexus#ip netns exec management ping test.cisco.com -c 5
PING test.cisco.com (192.168.2.100) 56(84) bytes of data.
64 bytes from test.cisco.com (192.168.2.100): icmp_seq=1 ttl=59 time=0.617 ms
64 bytes from test.cisco.com (192.168.2.100): icmp_seq=2 ttl=59 time=0.341 ms
64 bytes from test.cisco.com (192.168.2.100): icmp_seq=3 ttl=59 time=0.310 ms
64 bytes from test.cisco.com (192.168.2.100): icmp_seq=4 ttl=59 time=0.379 ms
64 bytes from test.cisco.com (192.168.2.100): icmp_seq=5 ttl=59 time=0.296 ms
```

```
--- test.cisco.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.296/0.388/0.617/0.119 ms
```

DNS 확인을 테스트하는 데 사용되는 네임스페이스에 인터넷에 대한 IP 연결이 있으면 **cisco.com**에 ping을 수행하여 내부 도메인 이름 외에도 외부 도메인 이름을 확인할 수 있는지 확인할 수 있습니다. 이는 공용 웹 서버에 대해 curl 및 wget과 같은 유틸리티를 사용해야 하는 경우 특히 중요합니다. 이 예에서는 관리 네임스페이스(인터넷에 IP 연결) 내에서 ping 유틸리티를 사용하여 올바른 외부 DNS 확인을 확인하는 방법을 보여 줍니다.

```
root@Nexus#ip netns exec management ping cisco.com -c 5
PING cisco.com (72.163.4.161) 56(84) bytes of data.
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=1 ttl=239 time=29.2 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=2 ttl=239 time=29.2 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=3 ttl=239 time=29.3 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=4 ttl=239 time=29.2 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=5 ttl=239 time=29.2 ms
```

```
--- cisco.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 29.261/29.283/29.335/0.111 ms
```

## /etc/resolv.conf 파일 형식

몇 가지 일반적인 컨피그레이션 매개변수에 대해 설명합니다. 환경에 맞게 모든 컨피그레이션 매개변수를 수정해야 합니다.

- **domain {domain-name.tld}** - 기본 도메인 이름 {domain-name.tld} 마침표로 끝나지 않은 호스트 이름에 추가합니다. /etc/resolv.conf 파일 내에 항목은 하나만 있을 수 있습니다.
- **search {domain-name-1.tld} [domain-name-2.tld...]** - 호스트 이름에 추가할 도메인 이름 ({domain-name-1.tld} 및 선택적으로 [domain-name-2.tld])을 공백으로 구분하여 정의합니다.

**참고:** 및 항목은 상호 배타적입니다. 한 번에 하나만 사용할 수 있습니다. 두 항목이 모두 /etc/resolv.conf 파일에 포함되어 있으면 파일에 마지막으로 나타나는 항목이 사용됩니다.

- **nameserver {address-1}** - DNS 확인 요청이 전달되는 DNS 서버에 대한 IP 주소 {address-1}을 정의합니다. 단일 파일 내에서 여러 서버 항목이 허용되며 최대 3개까지 허용됩니다.

## 예

이 예에서는 환경의 기본 도메인이 cisco.com이고 환경 내의 DNS 서버에 192.168.2.1 및 192.168.2.2의 IP 주소가 있는 /etc/resolv.conf 파일의 내용을 보여 줍니다. 이 시나리오에서 Bash 셸이 foo의 호스트 이름을 가진 디바이스의 IP 주소를 확인해야 하는 경우 **호스트**의 FDQN(Fully Qualified Domain Name)이 focisco.com인 호스트 이름의 끝에 cisco.com이 추가됩니다. 예

```
domain cisco.com
nameserver 192.168.2.1
nameserver 192.168.2.2
```

다음 예에서는 **cisco.com** 또는 **bar.com** 도메인 이름을 사용하여 DNS 호스트 이름을 확인할 수 있는 **/etc/resolv.conf** 파일의 내용을 보여줍니다. 환경 내의 DNS 서버에는 192.168.2.1 및 192.168.2.2의 IP 주소가 있습니다. 이 시나리오에서 Bash 셸이 foo의 호스트 이름을 가진 디바이스의 IP 주소를 확인해야 하는 경우 **foo.cisco.com**을 먼저 확인한 다음 **foo.cisco.com**에 대한 확인이 실패하면 다음 **foo.bar.com**을 확인하려고 시도합니다.

```
search cisco.com bar.com
nameserver 192.168.2.1
nameserver 192.168.2.2
```

## 관련 정보

- [Cisco Nexus 9000 Series NX-OS 프로그래밍 가이드, 릴리스 9.x](#)
- [Cisco Nexus 9000 Series NX-OS 프로그래밍 가이드, 릴리스 7.x](#)
- [Cisco Nexus 9000 Series NX-OS 프로그래밍 가이드, 릴리스 6.x](#)
- [Cisco Nexus 3000 Series NX-OS 프로그래밍 가이드, 릴리스 9.x](#)
- [Cisco Nexus 3000 Series NX-OS 프로그래밍 가이드, 릴리스 7.x](#)
- [Cisco Nexus 3000 Series NX-OS 프로그래밍 가이드, 릴리스 6.x](#)
- [Cisco Nexus 3500 Series NX-OS 프로그래밍 가이드, 릴리스 9.x](#)
- [Cisco Nexus 3500 Series NX-OS 프로그래밍 가이드, 릴리스 7.x](#)
- [Cisco Nexus 3500 Series NX-OS 프로그래밍 가이드, 릴리스 6.x](#)
- [Cisco Nexus 3600 Series NX-OS 프로그래밍 가이드, 릴리스 9.x](#)
- [Cisco Nexus 3600 Series NX-OS 프로그래밍 가이드, 릴리스 7.x](#)
- [Cisco Open NX-OS를 통한 프로그래밍 기능 및 자동화](#)