



Bash

- [About Bash, on page 1](#)
- [Accessing Bash, on page 1](#)
- [Escalate Privileges to Root, on page 2](#)
- [Examples of Bash Commands, on page 3](#)

About Bash

In addition to the NX-OS CLI, Cisco Nexus 9000 Series devices support access to the Bourne-Again SHell (Bash). Bash interprets commands that you enter or commands that are read from a shell script. Using Bash enables access to the underlying Linux system on the device and to manage the system.

Accessing Bash

In Cisco NX-OS, Bash is accessible from user accounts that are associated with the Cisco NX-OS dev-ops role or the Cisco NX-OS network-admin role.

The following example shows the authority of the dev-ops role and the network-admin role:

```
switch# show role name dev-ops

Role: dev-ops
  Description: Predefined system role for devops access. This role
               cannot be modified.
  Vlan policy: permit (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
-----
Rule    Perm    Type    Scope    Entity
-----
4       permit  command
3       permit  command
2       permit  command
1       permit  command
               conf t ; username *
               bcm module *
               run bash *
               python *

switch# show role name network-admin

Role: network-admin
  Description: Predefined network admin role has access to all commands
               on the switch
-----
```

Rule	Perm	Type	Scope	Entity
1	permit	read-write		

switch#

Bash is enabled by running the **feature bash-shell** command.

The **run bash** command loads Bash and begins at the home directory for the user.

The following examples show how to enable the Bash shell feature and how to run Bash.

```
switch# config t
switch(config)# feature bash-shell

switch# run?
run          Execute/run program
run-script   Run shell scripts

switch# run bash?
bash        Linux-bash

switch# run bash
bash-4.2$ whoami
admin
bash-4.2$ pwd
/bootflash/home/admin
bash-4.2$
```



Note

You can also execute Bash commands with the **run bash command** command.

The following is an example of the **run bash command** command.

```
run bash whoami
```

Escalate Privileges to Root

The privileges of an admin user can escalate their privileges for root access.

The following are guidelines for escalating privileges:

- admin privilege user (network-admin / vdc-admin) is equivalent of Linux root privilege user in NX-OS
- Only an authenticated admin user can escalate privileges to root, and password is not required for and authenticated admin privilege user.
- Bash must be enabled before escalating privileges.

The following example shows how to escalate privileges to root and how to verify the escalation:

```
switch# run bash
bash-4.2$ sudo su root
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.

```
#3) With great power comes great responsibility.

Password:

bash-4.2# whoami
root
bash-4.2# exit
exit
```

Examples of Bash Commands

This section contains examples of Bash commands and output.

Displaying System Statistics

The following example shows how to display system statistics:

```
switch# run bash
bash-4.2$ cat /proc/meminfo
<snip>
MemTotal:      16402560 kB
MemFree:       14098136 kB
Buffers:       11492 kB
Cached:        1287880 kB
SwapCached:    0 kB
Active:        1109448 kB
Inactive:      717036 kB
Active(anon):  817856 kB
Inactive(anon): 702880 kB
Active(file):  291592 kB
Inactive(file): 14156 kB
Unevictable:   0 kB
Mlocked:      0 kB
SwapTotal:     0 kB
SwapFree:      0 kB
Dirty:         32 kB
Writeback:     0 kB
AnonPages:     527088 kB
Mapped:        97832 kB
<\snip>
```

Running Bash from CLI

The following example shows how to run a bash command from the CLI with the **run bash command** command:

```
switch# run bash ps -el
F S    UID    PID    PPID    C  PRI   NI  ADDR  SZ  WCHAN  TTY          TIME CMD
4 S     0      1      0  0  80    0  -    528 poll_s ?        00:00:03 init
1 S     0      2      0  0  80    0  -     0 kthrea ?        00:00:00 kthreadd
1 S     0      3      2  0  80    0  -     0 run_ks ?        00:00:56 ksoftirqd/0
1 S     0      6      2  0 -40   -  -     0 cpu_st ?        00:00:00 migration/0
1 S     0      7      2  0 -40   -  -     0 watchd ?        00:00:00 watchdog/0
1 S     0      8      2  0 -40   -  -     0 cpu_st ?        00:00:00 migration/1
1 S     0      9      2  0  80    0  -     0 worker ?        00:00:00 kworker/1:0
1 S     0     10      2  0  80    0  -     0 run_ks ?        00:00:00 ksoftirqd/1
```

Running Python from Bash

The following example shows how to load Python and configure a switch using Python objects:

```
switch# run bash
bash-4.2$ python
Python 2.7.5 (default, Oct 8 2013, 23:59:43)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from cisco import *
>>> from cisco.vrf import *
>>> from cisco.interface import *
>>> vrfobj=VRF('myvrf')
>>> vrfobj.get_name()
'myvrf'
>>> vrfobj.add_interface('Ethernet1/3')
True
>>> intf=Interface('Ethernet1/3')
>>> print intf.config()

!Command: show running-config interface Ethernet1/3
!Time: Mon Nov 4 13:17:56 2013

version 6.1(2)I2(1)

interface Ethernet1/3
    vrf member myvrf

>>>
```