

Release 0.9 Giugno 2010

Rivisto Maggio 2017

Aggiunte Aprile 2018

## NX-OS

NX-OS e' il nuovo sistema operativo della Cisco utilizzato nelle piattaforme di fascia alta. Si tratta di un'evoluzione di IOS pensata per il datacenter. Al centro di NX-OS c'e' il concetto di virtualizzazione.

La virtualizzazione esiste da molti anni e nasce con le VLAN, che di fatto separavano logicamente un unico switch fisico. Ma le VLAN operano esclusivamente a layer 2. Successivamente, in anni piu' recenti, e' stato introdotto il concetto di Virtual Routing and Forwarding (VRF) che di fatto permette di creare dei router logici all'interno di un unico router fisico. Sostanzialmente opera a livello tre ed ogni VRF ha una sua tabella di routing e delle sue porte fisiche che operano indipendentemente dalle tabelle di routing e dalle porte degli altri VRF. Tutte le versioni di IOS consentono oramai l'uso di VRF.

La limitazione del VRF e' che non c'e' una separazione dei processi e delle risorse da loro utilizzate (processore, memoria) ma questo e' una conseguenza della non modularita' di IOS. Pertanto anche se ci sono diverse tabelle di routing e' c'e' una separazione logica di un unico router fisico di fatto un fault di un processo coinvolge tutti i VRF. Se va in crash il BGP o l'OSPF, per fare un esempio, il router non fornira' il servizio OSPF o BGP per tutti i VRF.

Con NX-OS e' stato fatto un ulteriore passo avanti. Si tratta di un sistema operativo modulare, dove ciascun processo gira nel suo spazio di memoria protetto. I processi si possono raggruppare ed associare a diversi contesti e a delle risorse hardware specifiche. Allora nel singolo device fisico abbiamo device logici completamente separati. Anche il management e' separato per ogni device logico. In NX-OS si hanno quelli che si chiamano "differenti **vdc**" ovvero "differenti **virtual device context**". Ogni vdc e' un gruppo di processi autonomi e indipendenti con proprie risorse con un proprio management. E questo e' il grande balzo in avanti rispetto al VRF.

E' un po' come una macchina virtuale con VMWARE. Il crash di un processo all'interno di un vdc non tocca gli altri vdc. Ad esempio due processi OSPF con lo stesso id in due differenti vdc non interagiscono minimamente tra loro. Anche le amministrazioni tra differenti vdc sono separate. L'amministratore di un vdc vede e puo' modificare solo le risorse hardware ad esso assegnate. Ad esempio non vedra' le porte e le vlan che non fanno parte del proprio vdc.

I datacenter con la network virtualization saranno molto piu' efficienti e funzionali. E' possibile avere differenti topologie logiche e una unica topologia fisica. Allora si potranno sfruttare al 100% tutte le risorse disponibili sovrapponendo reti logiche alla rete fisica

finche' ci saranno risorse sufficienti. Il cablaggio, il costo dell'hardware, il costo per il raffreddamento dei sistemi e il costo energetico sono solo alcuni benefici della network virtualization. Infine, ma non per importanza, il nuovo datacenter permette anche la "**fault isolation**" ovvero l'isolamento dei guasti all'interno della logical network in cui sono generati, e questo e' fondamentale per i clienti che utilizzano risorse.

Questo documento presenta rapidamente le nuove funzionalita' e le principali differenze tra IOS e NX-OS. La piattaforma di riferimento e' il Cisco Nexus 7000 che potete vedere in figura. Alla data di stesura di questo documento NX-OS e' arrivato alla release 5.0.

-----

### 1) "show module"

Come in IOS mostra i moduli hardware installati nello chassis.

Ecco esempio di output su Cisco Nexus 3064:

```
NEXUS3K-2# show module
Mod Ports          Module-Type          Model                Status
---  ---
1    64    48x10GE + 4x40G Supervisor  N3K-C3064PQ-10GX  active *

Mod Sw              Hw    Slot
---  ---
1    7.0(3)I7(2)      1.5   NA

Mod MAC-Address(es)  Serial-Num
---  ---
1    c0-8c-60-98-2d-48 to c0-8c-60-98-2d-87  IOC7380MV

Mod Online Diag Status
---  ---
1    Pass
```

\* this terminal session

```
NEXUS3K-2#
-----
```

### 2) "show module xbar"

Questi sono i moduli sul retro. Sono i moduli "fabric". Questo comando esiste nei modelli modulari.

-----

### 3) Moduli "N7K-M132XP-12", "N7K-SUP1"

Il primo e' un modulo Ethernet a 10Gbps

N7K - Nexus 7000

M1 - Multipurpose forward engine first generation

32 - Numero di porte nel modulo

X - Velocita' delle porte. In questo caso 10gigabit

P - Le schede fisiche a 10giga installate. In questo caso SFP+

1 - Fabric module prima generazione (e' nel retro dello chassis)

2 - Due canali di 40Gbps collegati con il fabric module  
(questo ad esempio e' "1" nel caso di moduli 10/100/1000)

Il secondo e' il modulo processore centrale, ovvero la scheda supervisor.

-----  
4) "show enviorment power"

Mostra la potenza massima che ciascun componente consuma, ovvero nelle condizioni peggiori.

Ad esempio nel caso di un Nexus 3064:

NEXUS3K-2# show environment power

Power Supply:

Voltage: 12 Volts

-----

PS	Model	Input Type	Power (Watts)	Current (Amps)	Status
8241	N2200-PAC-400W	AC	396.00	33.00	ok
8242	N2200-PAC-400W	AC	396.00	33.00	ok

-----

Mod	Model	Power Requested (Watts)	Current Requested (Amps)	Power Allocated (Watts)	Current Allocated (Amps)	Status
1	N3K-C3064PQ-10GX	336.00	28.00	336.00	28.00	powered-up

-----

Power Usage Summary:

Power Supply redundancy mode: Redundant

Power Supply redundancy operational mode: Redundant

Total Power Capacity 792.00 W

Power reserved for Supervisor(s) 336.00 W

Power currently used by Modules 0.00 W

-----  
Total Power Available 456.00 W

---

### 5) *"show ver"*

Come in IOS mostra la versione di NX-OS installata nel sistema e altre informazioni. Il sistema gira su Linux, e il processore normalmente e' Intel Xeon. Il sistema consiste di due immagini: "kickstart", "system". La prima e' il Linux, la seconda contiene il sistema NX-OS.

Ad esempio su un 3064:

```
...snip...
Software
  BIOS: version 4.5.0
  NXOS: version 7.0(3)I7(2)
  BIOS compile time: 11/09/2017
  NXOS image file is: bootflash:///nxos.7.0.3.I7.2.bin
  NXOS compile time: 11/22/2017 13:00:00 [11/22/2017 21:55:29]
...snip...
```

---

### 6) *"show running-config"*

Come in IOS mostra la configurazione

---

### 7) *INFO*

Il telnet di default e' disabilitato e nel sistema si entra in ssh

---

### 8) *Interfaccia di management*

#### **"interface mgmt0"**

E' l'interfaccia di management che conterra' l'indirizzo IP per la configurazione del sistema. Questa corrisponde ad una interfaccia 10/100/1000 nella supervisor. E' inserita in un VRF riservato di default. Questo si chiama "management VRF". Il motivo e' che in questo modo il management viene completamente isolato per consentire il management di sistema in qualsiasi condizione.

#### **"show vrf"**

Mostra il "default" e il "management" VRF

#### **"show vrf management interface"**

Mostra che di default l'interfaccia mgmt0 e' nel VRF "management"

Non e' possibile aggiungere altre interfacce nel VRF di management.

Se vogliamo fare un 'ping' nella rete di management dobbiamo utilizzare "ping IPADDRESS vrf management"

Ecco una tipica configurazione per la rete di management:

```
vrf context management
  ip route 0.0.0.0/0 10.0.0.1
interface mgmt0
  vrf member management
  ip address 10.0.1.8/16
```

Nota che non e' la stessa cosa di:

```
ip route 0.0.0.0/0 mgmt0 10.0.0.1 vrf management
```

che e' una configurazione di route leak per destinazione fuori dal vrf attuale

La sintassi IOS oltretutto sarebbe questa:

```
ip route vrf management 0.0.0.0 0.0.0.0 10.0.0.1
```

-----

## 9) *Fast-Ethernet e Access-list*

Nella configurazione tutte le interfacce hanno il nome di "Ethernet"  
Non esistono piu' le notazioni "FastEthernet", "GigaEthernet"

A differenza dei vari tipi di access-list presenti in IOS qui sono stati unificati.

Non esistono piu' "standard ACL", "extended ACL", "numbered ACL" ma il solo comando e' "ip access-list".

Non c'e' etherchannel, ma solo port-channel

-----

## 10) *Novita' nella CLI, "conf session", servizi*

- Il nome utente di default e' "admin" e l'accesso e' in ssh;
- Sono disponibili anche i comandi presenti in IOS solo in modalita' privilegiata. Ad esempio possiamo fare un 'ping';
- A differenza di IOS ritroviamo anche comandi specifici di linux, come "grep". Ad esempio "*show run | grep mgmt*";
- Oltre che "conf t" abbiamo ora "**conf session**". E' possibile scrivere una parte di configurazione che non sara' subito operativa, come accade normalmente in IOS. Ecco un esempio:

- a) "**conf session prova**"
- b) "**ip access-list prova**"  
" **permit tcp 10.15.20.20/24 any**"
- c) "**int e2/4**"  
"**ip access-group prova in**"

Non appena abbiamo terminato e verificato l'inserimento utilizziamo il comando "**verify**". Il sistema operativo verifica che ci sono risorse hardware sufficienti per quanto inserito

nella session. Se tutto e' ok possiamo mettere in esercizio le modifiche con "**commit**" all'interno della modalita' "config session" oppure "**abort**" nel caso in cui non vorriamo mettere in esercizio la session.

Ecco un esempio di configurazione per un utente ssh:

```
username admin password 5 $5$gnmTmN3UjiKk6 role network-admin
```

**Importante:** NX-OS e' un sistema modulare. Alcuni comandi potrebbero non essere disponibili se non si attiva il modulo associato.

Con il comando "**feature**" vediamo l'elenco delle feature e troviamo i protocolli di routing ad esempio, RIP, OSPF, BGP.

Con "**feature eigrp**" ad esempio attiviamo il modulo di eigrp. Dobbiamo ovviamente accertarsi di avere acquistato la licenza. Se non l'abbiamo il sistema ci avvisa ma ci permette di provarla per un certo periodo senza licenza.

-----  
11) "*roles*"

Con "**show role**" vedrete "vdc-admin" e "vdc-operator"

I ruoli servono per definire gruppi di utenti che hanno alcuni privilegi sul sistema.

I due ruoli di default, ad esempio, rappresentano un gruppo di "admin" ovvero amministratori,

in grado di fare qualsiasi cosa nel sistema, e un gruppo di "operatori", in grado di fare solo consultazioni, ovvero vedere ma non modificare la configurazione

Con "**show role feature**" vediamo un lungo elenco di attivita' che e' possibile associare ad un gruppo.

Ogni attivita' e' un permesso ad eseguire una certa operazione.

Con "**show role feature-group**" vediamo un elenco di gruppi di attivita', ovvero raccolte di attivita'.

Per esempio il feature group "L3" contiene le feature relative al bgp,ospf e cosi' via.

Creiamo un nuovo ruolo:

- a) "**role name gianrico**"
- b) "**rule 1 permit read**"
- c) "**rule 2 permit command ping \***"
- d) "**rule 3 permit command conf t : interface \***"
- e) "**interface policy deny**"
- f) "**permit interface ethernet 3/2**"

Questo ruolo permette l'uso del comando "ping" e la configurazione della sola interfaccia ethernet3/1.

Non e' possibile con IOS una tale granularita' nei permessi.

Con "**show role name gianrico**" verifichiamo la configurazione.

A questo punto creiamo un nuovo utente che ha questo ruolo:

**"username maurizio password prova role gianrico"**

Questo utente avra' solo i permessi che abbiamo messo nel ruolo.

---

## 12) "virtual-device"

Quando entriamo nel sistema come "admin" abbiamo il supporto per i **VDC (Virtual Device Context)**.

E' un concetto molto importante e innovativo. Di fatto sono delle macchine virtuali logicamente indipendenti.

Ogni macchina virtuale si puo' "ritagliare" una parte di chassis, ad esempio avere alcune interfacce.

**"show vdc"** <-- mostra le macchine virtuali attive

**"show vdc membership"** <--- fa vedere quali interfacce sono associate a quale vdc

**"show vdc NOME detail"**

**"show vdc NOME resources"**

**"vdc prova"** <-- da modalita' "conf t" crea un nuovo "vdc" di nome prova (anche per questi ci vogliono le licenze)

**"allocate interface e2/31-35"** <-- associamo queste interfacce al vdc corrente

**"ha-policy dual-sup bringdown/restart/switchover"** <-- queste sono tre opzioni. Se il VDC va in crash puo' supportare

o meno la ridondanza e utilizzarla se configurata.

E' possibile anche limitare le risorse in un VDC come il numero di VRF che si possono creare, il numero massimo di VLAN, il numero di port-channel

**"limit-resource ?"** <-- per limitare le risorse.

Una volta preparato per attivare il nuovo VDC si dovra' utilizzare il comando:

**"switchto vdc NOME"**

Ci sara' un wizard che chiedera' password, i dati della management interface, se di default il sistema e L2 o L3, e altro.

Massimo si possono usare 4 VLC (nella release della data del presente documento)

---

## 13) "rolling-back"

Una importante novita' in NX-OS e' la possibilita' di memorizzare lo stato del sistema e di ritornare a stati precedenti.

E' un qualcosa di simile a quello che abbiamo con i sistemi operativi. Ad esempio in Windows con il "ripristino" e i "punti di ripristino". Nel caso di NX-OS si ripristina il file di configurazione.

**"checkpoint prova"** <--- Crea un nuovo checkpoint di nome "prova"  
**"show checkpoint summary"** <--- Mostra i checkpoint presenti nel sistema  
**"show checkpoint prova"** <--- Mostra la configurazione di sistema nel  
 checkpoint "prova"  
**"rollback running-config checkpoint prova"** <--- Riporta la configurazione del sistema  
 a quella del checkpoint "prova"

Si possono creare fino a 10 checkpoint per ogni virtual-device (alla data del presente documento).

---

#### 14) OSPF

A differenza di IOS, dove si utilizza il comando "network" all'interno di "router OSPF" per stabilire quali interfacce partecipano al routing OSPF, in NX-OS all'interno di ogni singola interfaccia si deve inserire il comando **"ip router ospf PROCESSID area AREANUMBER"**.

**"show running-config ospf"** mostra la configurazione ospf senza necessita' di usare il pipe

---

#### 15) NX-OS, sistema operativo modulare

Essendo modulare, se un processo va in crash, il sistema NON va in crash. Tutti i protocolli layer 2 e OSPF sono "stateful process restart". Se OSPF va in crash il router adiacente non se ne accorge neppure. Il sistema lo riavvia e ne conserva lo stato precedente. Questa e' la "conservazione dello stato".

---

#### 16) Netflow

E' supportata la versione 5 e 9.

**"feature netflow"** <-- attiviamo il modulo, altrimenti non troveremo neanche i comandi CLI di configurazione netflow

**"flow record nxos"**

" match ipv4 destination address"

" match ipv4 source address"

" match ip protocol"

" collect transport tcp flags"

" collect routing forwarding-status" <-- invia info nel caso in cui un flusso e' stato bloccato, ad esempio per un ACL

**"flow exporter nxos-exp"**

" destination 172.16.10.10"

**"flow monitor miomonitor"**

" record nxos"  
" exporter nxos-exp"

"interface e3/2"  
" ip flow monitor miomonitor"

---

#### 17) *wireshark*

E' un pacchetto di sniffer molto diffuso. E' incluso in Nexus7000.

E' estremamente utile quando si fa' debugging avanzato. E' una funzionalita' inesistente in IOS dove si devono utilizzare necessariamente le SPAN ports oppure NETFLOW per poter fare lo stesso.

Con wireshark il debugging avanzato diventa molto piu' semplice e rapido.

"ethalyzer local interface mgmt brief limit-capture-frames 5" <-- Fa vedere 5 pacchetti di sniffer relativi all'interfaccia di mgmt

---

#### 18) *OTV*

OTV e' una nuova tecnologia messa a disposizione da Cisco sulla piattaforma Nexus7000 che consente l'integrazione di data-center geograficamente distanti per la creazione di un unico data-center logico. L'obiettivo della tecnologia e' quello di consentire in modo efficiente la fornitura di servizi da parte di piu' datacenter e la continuita' di servizio in caso di fault. Pertanto questa tecnologia e' ottimizzata per consentire la transizione e sincronizzazione di server virtuali tra datacenter e per fornire ai fornitori di servizi continuita' e efficienza in quanto permette di distribuire il carico efficacemente tra i datacenter. La feature e' introdotta con NX-OS 5.0.

I dati vengono trasportati come Layer2 e ogni Nexus in ogni datacenter mantiene una tabella di corrispondenza dei MAC per tutti i datacenter. I dati vengono trasportati tra i data-center utilizzando protocolli e tecnologie esistenti, come MPLS, IP e SONET/SDH.

---

#### 19) *vPC (Virtual Port Channel)*

vPC e' una tecnologia disponibile sulla piattaforma Nexus che sostituisce la VSS in uso sulla serie di switch Cisco 6500. Consente l'uso di due chassis in configurazione ridondata e quindi etherchannel di tipo multichassis tra switch di access e switch Nexus di core (nel caso di un two tier data-center) o di un server con due switch Nexus di core (nel caso di one tier data-center).

Per garantire la ridondanza sino a server e' necessario che essi supportino Etherchannel (con o senza LACP, ma LACP e' raccomandato)

Con vPC entrambi i Nexus sono nello stato di Active. Gli switch di access possono essere di qualsiasi produttore.

Tra i due Nexus ci dev'essere un link diretto chiamato vPC peer-link per permettere il funzionamento del vPC.

Quindi non si utilizza STP in quanto entrambi i link sono attivi e un server che supporta etherchannel puo' operare in Active-Active mode tra le sue schede di rete. Ovviamente se si vuole un comportamento Active/Standby si puo' utilizzare STP invece di vPC.

Da notare che con il Catalyst 6500 si ha il VSS per fare qualcosa di simile. I Nexus non supportano il VSS, rimangono due switch distinti, tuttavia si "fondono" a livello L2 permettendo appunto funzionalita' come il vPC

vPC, HSRP, VRRP, LACP sono disponibili in tutta la serie Nexus-7000 e non necessitano di licenza aggiuntiva ma disponibili in NX-OS di base.

vPC si usa anche per DCI (Data Center Interconnect). Sostanzialmente in ogni datacenter hai due Nexus in una configurazione dual-VPC (ovvero VPC da entrambi i lati)

Server ----> Access-Layer (2) ----> Aggregation Layer (2) ----> Core (2)

---

## 20) Virtual Extensible LAN (VXLAN) (nexus)

Sono dei tunnel L2 pensati per gli ambienti datacenter, e sono una evoluzione delle VLAN. Sono in grado di muoversi su una rete L2 e L3 preesistente (overlay) e utilizzano multicast per la comunicazione tra i nodi. Molto utili in ambienti cloud in cui bisogna isolare un gran numero di app e tenants (organizzazioni). Infatti una architettura cloud per sua natura e' multi-tenant. Ogni tenant ha la sua rete logica. Si puo' fare con le VLAN ma si ha il limite delle 4096. Il VNI (Vxlan network identifier) e' il numero di segmento.

- 1- Il tunnel e' definito da pacchetti UDP porta 4789;
- 2- Una VXLAN e tra due VXLAN tunnel endpoints (VTEPs);
- 3- Le VLAN arrivano a 4096 segmenti e VXLAN a 16 milioni (24 bit) di segmenti;
- 4- A differenza delle VLAN non usano STP: In contrast, VXLAN packets are transferred through the underlying network based on its Layer 3 header and can take complete advantage of Layer 3 routing, equal-cost multipath (ECMP) routing, and link aggregation protocols to use all available paths.

Oltre che nei modelli Nexus 9000 anche su ASR1000/9000 si possono gestire le VXLAN e quindi mettere in comunicazione VXLAN con VLAN.

## 21) Principali licenze disponibili

*Feature base licenses (valgono per l'intero device)*

BASE: vPC, HSRP, VRRP, LACP

LAN\_ENTERPRISE\_SERVICES\_PKG: OSPF, BGP, VxLAN

LAN\_ADVANCED\_SERVICES\_PKG (nexus 7000): VDC

---

## 22) Differenza tra le serie Nexus

**La serie 7000/7700** e' la piu' flessibile e va bene sia come LAN core che come datacenter. Le serie 3000/5000/9000 sono piu' datacenter oriented. La serie 7700 e' una versione avanzata della 7000.

- La serie 7000 e' leader per i datacenter a 10G. La 7700 per il datacenter 40G e oltre. 7700 non sostituisce i 7000.
- I moduli della serie 7700 non si possono usare nella 7000 e viceversa.
- La serie 7000 arriva a 550Gbps per slot mentre la 7700 arriva a 1.3Tbps per slot.
- La capacita' complessiva va dai 17.6 Tbps della 7000 ai 83Tbps della 7700.
- Nel 7700 le supervisor sono half-width card
- I software NX-OS sono gli stessi

**Che caratteristiche ha la serie 9000?** E' orientata al DataCenter. Perche'?

1) Perche' supporta la modalita' Application Centric Infrastructure (ACI). Solo la 9000 la supporta (e la 2000 in quanto sono unita' satellite);

2) Le card si possono far passare dalla modalita' NX-OS (standalone) alla modalita' ACI che utilizza un differente sistema operativo. Piu' switch si possono quindi accorpate ed essere gestiti da un server centrale APIC (controller) che e' un server con una API open.

Sostanzialmete il server centrale APIC gestisce gli switch. Poi Openstack o Vmware o HyperV hanno un plug-in cisco che fa in modo che l'utente possa configurarsi le macchine virtuali e la sua rete di macchine dall'interfaccia openstack:

Si tratta di SDN (software defined network) dove e' il software nei server che controlla la configurazione degli switch.

**Che caratteristiche ha la serie 5000?**

Sono switch di fascia piu' bassa. Sono L2. Sono di aggregazione. Quelli di livello piu' alto hanno un modulo L3 opzionale (5548)

**Un'altra differenza tra N7K e N9K:**

Seguono i protocolli per fondere piu' datacenter tra loro (ad esempio una rete meta' da un lato e meta' dall'altro con la stessa classe di IP):

OTV/LISP (multisite) - N7K -

mBGP/EVPN (site-to-site) - N9K - N7K

## 23) Verifica della release di sistema operativo.

boot nxos bootflash:/nxos.7.0.3.I7.2.bin

```

NEXUS3K-2# sh ver
Cisco Nexus Operating System (NX-OS) Software
...snip...
Software
  BIOS: version 4.5.0
  NXOS: version 7.0(3)I7(2)
  BIOS compile time: 11/09/2017
  NXOS image file is: bootflash:///nxos.7.0.3.I7.2.bin
  NXOS compile time: 11/22/2017 13:00:00 [11/22/2017 21:55:29]
Hardware
  cisco Nexus3000 C3064PQ Chassis
  Intel(R) Celeron(R) CPU P4505 @ 1.87GHz with 3902868 kB
of memory.
  Processor Board ID FOC17380VMV
  Device name: NEXUS3K-2
  bootflash: 1638000 kB
  usb1: 0 kB (expansion flash)
...snip...
  System version: 7.0(3)I7(2)

```

```

NEXUS3K-1# sh run
!Command: show running-config
!Time: Thu Apr 12 12:37:18 2018
version 7.0(3)I7(2)
hostname NEXUS3K-1
...

```

```

NEXUS3K-2# sh boot
sup-1
  NXOS variable = bootflash:/nxos.7.0.3.I7.2.bin
  Boot Variables on next reload:
  sup-1
  NXOS variable not set
NEXUS3K-2#

```

```

NEXUS3K-2# dir bootflash:
 4096 Mar 28 12:38:43 2018 .rpmstore/
 4096 Mar 28 12:39:08 2018 .swtam/
...snip...
 640 Oct 19 23:02:53 2013 license_FOC1737R0RA_6_1.lic
 4096 Mar 28 13:44:09 2018 logflash/
885554688 Mar 28 12:17:48 2018 nxos.7.0.3.I7.2.bin
 0 Mar 28 12:41:04 2018 platform-sdk.cmd
 4096 Apr 09 08:13:01 2018 scripts/
1024 Apr 10 14:52:07 2018 sprom_2_0_1
1024 Apr 10 14:52:07 2018 sprom_3_0_0
1024 Mar 28 12:23:55 2018 sprom_cstruct_2_0_0
1024 Mar 28 12:24:29 2018 sprom_cstruct_3_0_0
 4096 Jan 29 22:58:27 2010 vdc_2/
 4096 Jan 29 22:58:27 2010 vdc_3/
 4096 Jan 29 22:58:27 2010 vdc_4/
 4096 Mar 28 12:24:13 2018 virtual-instance/

```

```

Usage for bootflash://sup-local

```

```
941039616 bytes used
709865472 bytes free
1650905088 bytes total
```

#### 24) Abilitazione per accesso SSH con cifratura debole

Il messaggio di errore e' il seguente:

```
NEXUS3K-2# 2018 Apr 12 14:23:20 NEXUS3K-2 %$ VDC-1 %$ %DAEMON-2-
SYSTEM_MSG: fatal: No matching ciphers found. Client (10.0.1.195)
supported ciphers : aes256-cbc,aes192-cbc,aes128-cbc,twofish-
cbc,blowfish-cbc,3des-cbc,arcfour . Server supported ciphers :
aes128-ctr,aes192-ctr,aes256-ctr - dcos_sshd[17189]
```

la soluzione e' questa:

```
NEXUS3K-2#
NEXUS3K-2# conf t
Enter configuration commands, one per line. End with CNTL/Z.
NEXUS3K-2(config)# ssh ci
cipher-mode    ciphers
NEXUS3K-2(config)# ssh cipher-mode weak
NEXUS3K-2(config)# exit
NEXUS3K-2#
```

#### 25) Accesso alla shell del sistema operativo:

```
NEXUS3K-2# run guestshell
Note: Guest shell is not enabled by default, use the 'guestshell
enable' command to enable it
```

```
NEXUS3K-2# gu
guestshell  gunzip
```

```
NEXUS3K-2# guestshell enable
NEXUS3K-2# 2018 Apr 12 14:26:06
NEXUS3K-2 %$ VDC-1 %$ %VMAN-2-INSTALL_STATE: Installing virtual
service 'guestshell+'
NEXUS3K-2# guestshell 2018 Apr 12 14:27:06
NEXUS3K-2 %$ VDC-1 %$ %VMAN-2-INSTALL_STATE: Install success
virtual service 'guestshell+'; Activating
2018 Apr 12 14:27:06
NEXUS3K-2 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Activating virtual
service 'guestshell+'
NEXUS3K-2 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully
activated virtual service 'guestshell+'
2018 Apr 12 14:28:21
NEXUS3K-2 %$ VDC-1 %$ %VMAN-2-GUESTSHELL_ENABLED: The guest shell
has been enabled. The command 'guestshell' may be used to access
it, 'guestshell destroy' to remove it.
```

```
[admin@guestshell ~]$
```

```
[admin@guestshell ~]$ ls
```

```
[admin@guestshell ~]$ cd ..
```

```
[admin@guestshell home]$ ls
```

```
admin guestshell
```

```
[admin@guestshell home]$ cd /
```

```
[admin@guestshell /]$ ls
```

```
bin                dev    lib64              proc          selinux
var
boot               etc    lost+found        root          srv
volatile
bootflash         home  media             rr_moved     sys
cisco             isan  mnt              run          tmp
created.2017-05-23.16:30:23 lib  opt              sbin         usr
[admin@guestshell /]$
```

```
[admin@guestshell /]$ uname -r
3.4.91-WR5.0.1.13_standard+
```

```
[admin@guestshell /]$ df
```

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/loop8	173533	145895	27138	85%	/
none	2097152	546464	1550688	27%	/cisco
none	614400	0	614400	0%	/cisco/core
none	5120	2108	3012	42%	/etc/shared
none	122880	4384	118496	4%	/var/nginx_local
none	102400	47780	54620	47%	/var/run/netns
none	614400	0	614400	0%	/volatile
/dev/sda3	1612212	1098432	513780	69%	/bootflash
/dev	1951436	0	1951436	0%	/dev
none	2097152	546464	1550688	27%	/cisco
none	614400	0	614400	0%	/cisco/core
none	5120	2108	3012	42%	/etc/shared
none	122880	4384	118496	4%	/var/nginx_local
none	102400	47780	54620	47%	/var/run/netns
none	614400	0	614400	0%	/volatile
none	614400	0	614400	0%	/cisco/core
tmpfs	1951436	0	1951436	0%	/dev/shm
tmpfs	1951436	8208	1943228	1%	/run
tmpfs	1951436	8	1951428	1%	/sys/fs/cgroup
none	1951436	24	1951412	1%	/var/volatile

```
[admin@guestshell /]$ ps -xa
```

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:00	/sbin/init
9	?	Ss	0:00	/usr/lib/systemd/systemd-journald
17	?	Ss1	0:00	/usr/sbin/rsyslogd -n
23	?	Ss	0:00	/usr/lib/systemd/systemd-logind
26	?	Ss	0:00	/bin/dbus-daemon --system --address=systemd: --nofork

```
32 ?          Ss      0:00 /usr/sbin/crond -n
33 pts/0      Ss+    0:00 /sbin/agetty --noclear ttyS0
34 pts/1      Ss+    0:00 /sbin/agetty --noclear ttyS1
74 ?          Ss      0:00 /usr/sbin/sshd -D -f /etc/ssh/
sshd_config-cisco -p 17
116 ?         Ss      0:00 sshd: admin [priv]
118 ?         S       0:00 sshd: admin@pts/4
119 pts/4      Ss      0:00 bash -li
155 pts/4     R+     0:00 ps -xa
```

```
[admin@guestshell ~]$ sudo su
[root@guestshell ~]#
```

```
[root@guestshell ~]# perl
bash: perl: command not found
```

```
[root@guestshell ~]# php
bash: php: command not found
```

```
[root@guestshell ~]# python
Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

```
[root@guestshell ~]# yum update
Loaded plugins: fastestmirror
Could not retrieve mirrorlist http://mirrorlist.centos.org/?
release=7&arch=x86_64&repo=os&infra=stock error was
14: curl#6 - "could not resolve host: mirrorlist.centos.org;
Unknown error"
```

...

## Copyright and General Advice Disclaimer

---

Tutti i marchi riportati appartengono ai legittimi proprietari; questo documento non è sponsorizzato o sottoscritto dalle società eventualmente citate. L'autore di questo documento non si assume nessuna responsabilità e non dà nessuna garanzia riguardante l'accuratezza o la completezza delle informazioni presenti nonché da conseguenze sull'uso delle informazioni presenti.

Copyright 2018 Gianrico Fichera

Nessuna parte di questa pubblicazione può essere riprodotta o trasmessa, in qualsiasi forma o con qualsiasi mezzo, elettronico, meccanico, fotocopie, registrazione, senza il consenso dell'autore al di fuori della disciplina del fair-use.

This material is not sponsored by, endorsed by, or affiliated with anyone. All trademarks are trademarks of their respective owners. I tried to ensure the accuracy and completeness of the contents of this document, but I cannot offer any undertaking or guarantee, either expressly or implicitly regarding how correct, complete or up to date the contents of this document are. I reserve the right to supplement this document at any time or to change or delete any information contained or views expressed.