



## **Cisco Nexus 9000v (9300v/9500v) ガイド、リリース 10.6(x)**

最終更新：2026 年 2 月 3 日

### **シスコシステムズ合同会社**

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】シスコ製品をご使用になる前に、安全上の注意（[www.cisco.com/jp/go/safety\\_warning/](http://www.cisco.com/jp/go/safety_warning/)）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: <https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html>. Cisco product warranty information is available at <https://www.cisco.com/c/en/us/products/warranty-listing.html>. US Federal Communications Commission Notices are found here <https://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and-if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2025 Cisco Systems, Inc. All rights reserved.



## 目次

### Trademarks ?

はじめに :

はじめに vii

対象読者 vii

表記法 vii

Cisco Nexus 9000 シリーズ スイッチの関連資料 viii

マニュアルに関するフィードバック viii

通信、サービス、およびその他の情報 ix

Cisco バグ検索ツール ix

マニュアルに関するフィードバック ix

第 1 章

新機能と更新情報 1

新機能と更新情報 1

第 2 章

概要 3

Cisco Nexus 9000v プラットフォーム ファミリの概要 3

ホストの最小メモリ要件 3

Cisco Nexus 9300v プラットフォーム 4

Cisco Nexus 9300v プラットフォーム コンポーネント 5

Cisco Nexus 9300v シャーシ 6

Cisco Nexus 9300v ライン カード 7

vNIC マッピング 7

Cisco Nexus 9500v プラットフォーム 8

Cisco Nexus 9500v プラットフォーム コンポーネント 10

Cisco Nexus 9500v シャーシ	11
Cisco Nexus 9500v ライン カード	12
vNIC マッピング	13
Nexus 9000v のスループット	15
Nexus 9000v 機能のサポート	16
レイヤ 2 の機能	17
レイヤ 3 の機能	17
VXLAN およびセグメント ルーティング機能	18
プログラマビリティ機能	19
Guestshell のサポート	20
Nexus 9000v Platform MIB のサポート	20
Nexus 9000v プラットフォームの注意事項と制限事項	21

---

### 第 3 章

<b>Nexus 9000v の展開</b>	<b>23</b>
Nexus 9000v ハイパーバイザ サポート	23
KVM/QEMU 属性	23
ESXi 属性	24
VirtualBox 属性	25
KVM/QEMU に対する Nexus 9000v 展開ワークフロー	25
共通展開ワークフロー	25
プラットフォーム特有のワークフロー	28
プラットフォームのインターコネクト	28
ESXi の Nexus 9000v 展開ワークフロー	31
共通展開ワークフロー	31
プラットフォーム特有のワークフロー	32
プラットフォームのインターコネクト	33
Vagrant 用 Nexus 9000v 展開ワークフロー	33
共通展開ワークフロー	34
プラットフォーム特有のワークフロー	34
Vagrant での Sync フォルダのサポート	34
プラットフォームのインターコネクト	37

イメージアップグレードのワークフロー	38
新しいアーティファクトからの展開	38
新しい NX-OS イメージからのアップグレード	39

---

## 第 4 章

<b>Nexus 9300v および 9500v Lite NX-OS イメージ</b>	<b>41</b>
Nexus 9300v および 9500v Lite NX-OS イメージについて	41
リソース要件	41
仮想アーティファクト	42
サポートされている機能	43
Nexus 9300v Lite および 9500v Lite NX-OS イメージの展開	45

---

## 付録 A :

<b>Cisco Nexus 9000v のトラブルシューティング</b>	<b>47</b>
Cisco Nexus 9000v プラットフォームのトラブルシューティング	47
すべてのハイパーバイザに共通の問題	48
ESXi ハイパーバイザの問題	49
KVM/QEMU ハイパーバイザの問題	51
Vagrant/VirtualBox の問題	51
Cisco Nexus 9000v データプレーンのトラブルシューティング	53
debug コマンド	53
Event History コマンド	54
コマンドの表示	54
パケット キャプチャ コマンド	57





## はじめに

この前書きは、次の項で構成されています。

- [対象読者](#) (vii ページ)
- [表記法](#) (vii ページ)
- [Cisco Nexus 9000 シリーズ スイッチの関連資料](#) (viii ページ)
- [マニュアルに関するフィードバック](#) (viii ページ)
- [通信、サービス、およびその他の情報](#) (ix ページ)

## 対象読者

このマニュアルは、Cisco Nexus スイッチの設置、設定、および維持に携わるネットワーク管理者を対象としています。

## 表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
<b>bold</b>	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
<i>italic</i>	イタリック体の文字は、ユーザが値を指定する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角かっこで囲んで示しています。
[x   y]	いずれか1つを選択できる省略可能なキーワードや引数は、角かっこで囲み、縦棒で区切って示しています。
{x   y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波かっこで囲み、縦棒で区切って示しています。

表記法	説明
[x {y   z}]	角かっこまたは波かっこが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角かっこ内の波かっこと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
variable	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しないでください。引用符を使用すると、その引用符も含めて string と見なされます。

例では、次の表記法を使用しています。

表記法	説明
screen フォント	スイッチが表示する端末セッションおよび情報は、スクリーンフォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の screen フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ (<>) で囲んで示しています。
[]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符 (!) またはポンド記号 (#) がある場合には、コメント行であることを示します。

## Cisco Nexus 9000 シリーズ スイッチの関連資料

Cisco Nexus 9000 シリーズ スイッチ全体のマニュアルセットは、次の URL にあります。

[https://www.cisco.com/en/US/products/ps13386/tsd\\_products\\_support\\_series\\_home.html](https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html)

## マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

## 通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、[Cisco Profile Manager](#) でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、[Cisco Services](#) にアクセスしてください。
- サービス リクエストを送信するには、[Cisco Support](#) にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、[Cisco DevNet \[英語\]](#) にアクセスしてください。
- 一般的なネットワーキング、トレーニング、認定関連の出版物を入手するには、[Cisco Press](#) にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、[Cisco Warranty Finder](#) にアクセスしてください。

## Cisco バグ検索ツール

[シスコバグ検索ツール](#) (BST) は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理するシスコバグ追跡システムへのゲートウェイです。BSTは、製品とソフトウェアに関する詳細な障害情報を提供します。

## マニュアルに関するフィードバック

シスコのテクニカルドキュメントに関するフィードバックを提供するには、それぞれのオンラインドキュメントの右側のペインにあるフィードバックフォームを使用してください。





# 第 1 章

## 新機能と更新情報

- [新機能と更新情報（1 ページ）](#)

## 新機能と更新情報

この表は、リリース 10.6(x) で導入または強化された機能のリストを示しています。

表 1: 10.6(x) の新機能および変更情報

特長	説明	変更が行われたリリース	参照先
該当なし	このリリースで追加された新機能はありません。	10.6(1)F	該当なし





## 第 2 章

### 概要

この章は、次の項で構成されています。

- [Cisco Nexus 9000v プラットフォーム ファミリの概要 \(3 ページ\)](#)
- [Cisco Nexus 9300v プラットフォーム \(4 ページ\)](#)
- [Cisco Nexus 9500v プラットフォーム \(8 ページ\)](#)
- [Nexus 9000v のスループット \(15 ページ\)](#)
- [Nexus 9000v 機能のサポート \(16 ページ\)](#)
- [Nexus 9000v Platform MIB のサポート \(20 ページ\)](#)
- [Nexus 9000v プラットフォームの注意事項と制限事項 \(21 ページ\)](#)

## Cisco Nexus 9000v プラットフォーム ファミリの概要

Cisco Nexus 9000v は、Cisco Nexus 9000 ソフトウェアを実行するスタンドアロン スイッチのコントロールプレーンの側面をシミュレートするように設計された仮想プラットフォーム ファミリです。このプラットフォーム ファミリは、Cisco Nexus 9000 ハードウェア プラットフォームを実行するのと同じソフトウェア イメージを使用します。仮想プラットフォームは特定の ASIC またはハードウェア SKU をシミュレートしようとはしませんが、それらに対応するハードウェアと連携しています。最適化されたシスコ ソフトウェア データ プレーンは、ラインカード インターフェイス全体のトラフィックを処理します。Cisco Nexus 9000v 仮想プラットフォーム ファミリは、Nexus 9300v と Nexus 9500v の 2 つの仮想プラットフォームで構成されています。次のセクションでは、これら 2 つのプラットフォームの機能について説明します。

Nexus 9000v プラットフォーム ファミリの仮想プラットフォームを使用すると、費用対効果の高い方法でネットワークをシミュレートできます。シミュレートされたネットワークを使用して、実稼働ネットワークに適用する前に構成を検証します。これらのプラットフォームを使用して、Cisco NX-OS プログラマビリティ インターフェイスを使用してネットワーク自動化ツールを迅速に開発およびテストします。

## ホストの最小メモリ要件

Nexus 9000v を起動するには、最低 10GB の RAM が必要です。基礎となるホスト (またはラップトップ) に使用可能な追加のメモリがあることを確認してください。

# Cisco Nexus 9300v プラットフォーム

Cisco Nexus 9300v プラットフォームは、単一の併置されたラインカードを備えた単一のスーパーバイザの非モジュラシャーシをシミュレートします。この仮想シャーシは、「lxc」モードで実行されるスタンドアロンの Cisco Nexus 9300 ハードウェアプラットフォームと密接に連携しています。次の表は、この仮想プラットフォームの仕様を示しています。

## フォームファクタ

コンポーネント/パラメータ	仕様
使用方法	シミュレーション
バイナリ	NX-OS ハードウェアと同じ
管理インターフェイス	1
ラインカード	1
ラインカードインターフェイス	64

## リソース要件

Resource	仕様
最小構成の RAM	10.0G（基本ブートアップ）
推奨 RAM	12.0 G（機能の数による）
最小 vCPU	4
推奨される vCPUs	4
最小 vNIC	1
Maximum vNICs	65

## 展開環境

- KVM/QEMU 4.2.0
- ESXI 8.0
- Vagrant 2.3.7

Nexus 9300v プラットフォームを展開するには、Cisco CCO から適切な仮想アーティファクトを取得します。次の表に、サポートされている仮想アーティファクトを示します。仮想マシンを展開すると、それ自体が Nexus 9300v プラットフォームであると報告されます。



- (注) Cisco Nexus 9300v プラットフォームは、Cisco NX-OS リリース 10.1 (1) まで 32 ビット イメージのみをサポートしました。Cisco NX-OS リリース 10.2 (1) 以降、64 ビット イメージのみがサポートされます。

次の表に、仮想アーティファクトを示します。

ハイパーバイザ	仮想アーティファクト	説明
ESXI 8.0	nexus9300v.10.1.1.ova	仮想ディスク、マシン定義、および NXOS イメージが含まれています。
KVM/QEMU 4.2.0	nexus9300v.10.1.1.qcow2	ブートフラッシュに仮想ディスクと NXOS イメージが含まれています。
Vagrant 2.3.7	nexus9300v.10.1.1.box	マシン定義とともに仮想ディスクにプレインストールされた NXOS イメージが含まれています。

最初の仮想マシンの展開後、一般的な NX-OS ワークフロー（例：install all <>）を使用して、プラットフォーム上の Cisco NX-OS イメージをアップグレードできます。



- (注) 古い Nexus 9000v を現在のリリースにアップグレードすると、自動的に Nexus 9300v に変換されます。その後のリロードと NX-OS イメージのアップグレードの後でも、プラットフォームは引き続き Nexus 9300v として表示されます。

## Cisco Nexus 9300v プラットフォーム コンポーネント

Cisco Nexus 9300v プラットフォームは、参照ハードウェアの対応物と同様に、シャーシ、スーパーバイザ、および 1 つのラインカードの 3 つの主要コンポーネントで構成されています。下表は製品 ID (PID) およびプラットフォームに関連付けられた SNMP sysOID を示します

コンポーネント	説明	PID	sysOID
シャーシ	Nexus9000 C9300v シャーシ	N9K-C9300v	EntPhysicalVendorType =cevChassisN9KV9000
Supervisor (スーパーバイザ モジュール)	スーパーバイザ モ ジュール	N9K-vSUP	(注) 下位互換性を確保する ために、以前のリリー スの Nexus 9000v プ ラットフォームで使用 されていた sysOID が Nexus 9300v プラット フォームに再利用され ています。
ラインカード	Nexus 9000v 64 ポート イーサネットモジュール	N9K-X9364v	

## Cisco Nexus 9300v シャーシ

以下は、関連する **show** コマンドからのシャーシ関連情報の出力例を示しています。

```
switch# sh version
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html
Copyright (c) 2002-2022, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained herein are owned by
other third parties and are used and distributed under license.
Some parts of this software are covered under the GNU Public
License. A copy of the license is available at
http://www.gnu.org/licenses/gpl.html.

Nexus 9000v is a demo version of the Nexus Operating System

Software
  BIOS: version
  NXOS: version 10.2(3) [build 10.2(2.185)] [Feature Release]
  BIOS compile time:
  NXOS image file is: bootflash:///nxos64-cs.10.2.2.185.F.bin
  NXOS compile time: 3/30/2022 13:00:00 [03/31/2022 00:30:59]

Hardware
  cisco Nexus9000 C9300v Chassis
  Intel(R) Xeon(R) CPU E5-2658 v4 @ 2.30GHz with 20499656 kB of memory.
  Processor Board ID 9GFDLI2JD0R
  Device name: switch
  bootflash: 4287040 kB

Kernel uptime is 1 day(s), 23 hour(s), 35 minute(s), 21 second(s)

Last reset
Reason: Unknown
System version:
Service:

plugin
Core Plugin, Ethernet Plugin

Active Package(s):
```

```

switch#
switch# sh module
Mod Ports Module-Type Model Status
-----
1 64 Nexus 9000v 64 port Ethernet Module N9K-X9364v ok
27 0 Virtual Supervisor Module N9K-vSUP active *

Mod Sw Hw Slot
-----
1 10.2(2.185) 0.0 LC1
27 NA 0.0 SUP1

Mod MAC-Address(es) Serial-Num
-----
1 00-ed-c6-a2-01-01 to 00-ed-c6-a2-01-40 9JAUkW2T51G
27 00-ed-c6-a2-1b-01 to 00-ed-c6-a2-1b-12 9GFDLI2JD0R

Mod Online Diag Status
-----
1 Pass
27 Pass

* this terminal session
switch#
switch# sh inventory
NAME: "Chassis", DESCR: "Nexus9000 C9300v Chassis"
PID: N9K-C9300v , VID: , SN: 9ZQKP299FIZ

NAME: "Slot 1", DESCR: "Nexus 9000v 64 port Ethernet Module"
PID: N9K-X9364v , VID: , SN: 9JAUkW2T51G

NAME: "Slot 27", DESCR: "Supervisor Module"
PID: N9K-vSUP , VID: , SN: 9GFDLI2JD0R

switch#

```

## Cisco Nexus 9300v ライン カード

Cisco Nexus 9300v プラットフォームは、64 個の仮想インターフェイスを備えた単一の仮想ラインカードをサポートします。プラットフォームが起動すると、ラインカードが自動的に装着されます。このシャーシにラインカードを挿入したり、取り外したりすることはできません。The ラインカード起動プロセスは、スーパーバイザがうまく起動し、「アクティブ」な状態になった後で開始します。対応するハードウェアと同様に、ラインカードの起動は「present」状態から開始され、「OK」状態に達すると完全に機能します。

## vNIC マッピング

実際の Cisco Nexus 9300 ハードウェアプラットフォームでは、ラインカードの前面パネルポートに光ファイバを「プラグイン」できます。Nexus 9300v などの仮想プラットフォームでは、必要な数の仮想ネットワーク インターフェイス カード/インターフェイス (vNIC) をハイパーバイザから Nexus 9300v プラットフォームにエクスポートする必要があります。

Nexus 9300v プラットフォームは、順次 vNIC マッピングを使用します。ハイパーバイザによって渡された最初の vNIC を Nexus 9300v 管理ポートにマッピングします。後続の vNIC は、ラインカードインターフェイスに順次マッピングされます。たとえば、2 つの vNIC を Nexus 9300v

にエクスポートすると、最初の vNIC が NX-OS 「mgmt」 インターフェイスにマッピングされます。2 つ目の vNIC は「イーサネット 1/1」 インターフェイスにマップされます。

## vNIC Mapping Informational Show コマンド

### Show Platform vNIC コマンド

Cisco Nexus 9300v プラットフォームでは、CLI コマンドを使用して、現在の vNIC マッピングスキーム、マッピングされている vNIC の数、および MAC アドレスから vNIC へのマッピングを表示できます。これらのコマンドを使用すると、正しい数の vNIC が仮想マシンに渡されたことを確認でき、どのインターフェースがマップされているかを確認できます。

show vNIC platform コマンドの出力例:

#### show platform vnic mapped

```
v-switch# show platform vnic mapped
      NXOS Interface      VNIC MAC-Address      Internal VNIC
      -----
Ethernet1/1      00c0.c000.0101      phyEth1-1
Ethernet1/2      00c0.c000.0102      phyEth1-2
Ethernet1/3      00c0.c000.0103      phyEth1-3
Ethernet1/4      00c0.c000.0104      phyEth1-4
Ethernet1/5      00c0.c000.0105      phyEth1-5
Ethernet1/6      00c0.c000.0106      phyEth1-6
Ethernet1/7      00c0.c000.0107      phyEth1-7
Ethernet1/8      00c0.c000.0108      phyEth1-8
Ethernet1/9      00c0.c000.0109      phyEth1-9
Ethernet1/10     00c0.c000.010a      phyEth1-10
Ethernet1/11     00c0.c000.010b      phyEth1-11
Ethernet1/12     00c0.c000.010c      phyEth1-12
Ethernet1/13     00c0.c000.010d      phyEth1-13
Ethernet1/14     00c0.c000.010e      phyEth1-14
Ethernet1/15     00c0.c000.010f      phyEth1-15
Ethernet1/16     00c0.c000.0110      phyEth1-16
```

#### show platform vnic info

```
v-switch# show platform vnic info
VNIC Scheme: Sequential
  mgmt0 interface: eth1 (00c0.c000.aabb)
  Module      # VNICs Mapped
  -----
  16          16
  -----
VNICs passed: 16
VNICs mapped: 16
VNICs unmapped: 0
```

## Cisco Nexus 9500v プラットフォーム

Cisco Nexus 9500v は、動的なライン カードの挿入と取り外しをサポートする、単一スーパーバイザプラットフォームの 16 スロット モジュラ シャーシをシミュレートします。この仮想シャーシは、スタンドアロンの Cisco Nexus 9500 ハードウェアプラットフォームと密接に連携しています。現在、このバージョンの Nexus 9500v は、モジュラ ハードウェア シャーシで通

常見られるシステムコントローラーまたはファブリックカードをシミュレートしていません。このプラットフォームは、汎用ラインカードの4つの異なるフォームファクタをサポートします。これらのラインカードは同じLinuxカーネルを共有し、サポートされるインターフェイスの数だけが異なります。次の表は、この仮想プラットフォームの仕様を示しています。

#### フォームファクタ

コンポーネント/パラメータ	仕様
使用方法	シミュレーション
バイナリ	NX-OS ハードウェアと同じ
管理インターフェイス	1
ラインカード	最大 16 個
ラインカードインターフェイス	KVM/QEMU 環境で最大 400 のインターフェイス

#### リソース要件

Resource	仕様
最小構成の RAM	10.0G (1つのラインカードでの基本的なブートアップ、追加のラインカードごとに 1.2G)
推奨 RAM	12.0G (機能数による)
最大 vCPU 数	4 (16枚のラインカードを構成する場合は、6個のvCPUをお勧めします)
最小 vNIC	1
Maximum vNICs	KVM/QEMU 環境の 400 個のインターフェイス

#### 展開環境

- KVM/QEMU 4.2.0
- ESXI 8.0

Cisco Nexus 9500v プラットフォームを展開するには、Cisco CCO から適切な仮想アーティファクトを取得します。下表はサポートされている仮想アーティファクトを示します。仮想マシンを展開すると、それ自体が Nexus 9500v として報告されます。



(注) Cisco Nexus 9500v プラットフォームは、Cisco NX-OS リリース 10.1(1) の 64 ビット イメージのみをサポートします。

テーブルには、仮想アーティファクトが表示されます。

ハイパーバイザ	仮想アーティファクト	説明
ESXI 8.0	nexus9500v64.10.1.1.ova	仮想ディスク、マシン定義、および NXOS イメージが含まれています  64-bit.ova ファイルは N9500v プラットフォームを起動し、次に Cisco NX-OS リリース 10.1(1) ソフトウェアの 64 ビット イメージを起動します。  (注) スーパーバイザは 64 ビットで、ラインカードは 32 ビットです。
KVM/QEMU 4.2.0	nexus9500v64.10.1.1.ova	ブートフラッシュの仮想ディスクと NXOS イメージを含みます。

最初の仮想マシンの展開後、一般的な NX-OS ワークフロー（例: **install all <>**）を使用して、プラットフォーム上の Cisco NX-OS イメージをアップグレードできます。



(注) 以前のリリースの Cisco Nexus 9000v を Nexus 9500v プラットフォームに変換することはできません。Nexus 9500v 仮想アーティファクトを適用することで変更できます。

## Cisco Nexus 9500v プラットフォーム コンポーネント

Cisco Nexus 9500v プラットフォームは、参照ハードウェアの対応物と同様に、シャーシ、スーパーバイザ、およびラインカードの3つの主要コンポーネントで構成されています。下表は製品 ID (PID) およびプラットフォームに関連付けられた SNMP sysOID を示します

コンポーネント	説明	PID	sysOID
シャーシ	Nexus9000 C9500v シャーシ	N9K-C9500v	EntPhysicalVendorType = cevChassisN9KC9500v
Supervisor (スーパーバイザ)	スーパーバイザ モジュール	N9K-vSUP	
ラインカード	Nexus 9000v 64 ポートイーサネットモジュール	N9K-X9564TX	

## Cisco Nexus 9500v シャーシ

次の参照は、単一のラインカードを備えた Nexus 9500v プラットフォームの関連の **show** コマンドからのサンプル シャーシ関連情報の出力です。

```
switch# sh version
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html
Copyright (c) 2002-2022, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained herein are owned by
other third parties and are used and distributed under license.
Some parts of this software are covered under the GNU Public
License. A copy of the license is available at
http://www.gnu.org/licenses/gpl.html.
```

Nexus 9000v is a demo version of the Nexus Operating System

```
Software
BIOS: version
NXOS: version 10.2(3) [build 10.2(2.191)] [Feature Release]
BIOS compile time:
NXOS image file is: bootflash:///nxos64-cs.10.2.2.191.F.bin
NXOS compile time: 4/5/2022 11:00:00 [04/05/2022 22:45:26]
```

```
Hardware
ciscoNexus9000 C9500v Chassis ("Supervisor Module")
Intel(R) Xeon(R) CPU E5-2658 v4 @ 2.30GHz with 16395468 kB of memory.
```

```
...skipping 1 line
Device name: switch
bootflash: 4287040 kB
```

Kernel uptime is 0 day(s), 0 hour(s), 7 minute(s), 51 second(s)

```
Last reset
Reason: Unknown
System version:
Service:
```

```
plugin
Core Plugin, Ethernet Plugin
```

Active Package(s):

```
switch#
switch# sh module
Mod Ports Module-Type Model Status
-----
1 64 Nexus 9000v 64 port Ethernet Module N9K-X9564v ok
27 0 Virtual Supervisor Module N9K-vSUP active *
```

```
Mod Sw Hw Slot
-----
1 10.2(2.191) 0.0 LC1
27 10.2(2.191) 0.0 SUP1
```

```
Mod MAC-Address(es) Serial-Num
-----
1 00-66-1e-aa-01-01 to 00-66-1e-aa-01-40 9LKOT3OP6A5
27 00-66-1e-aa-1b-01 to 00-66-1e-aa-1b-12 9V3ZW14YP0M
```

```

Mod Online Diag Status
---
1 Pass
27 Pass

* this terminal session
switch# sh inventory
NAME: "Chassis", DESCR: "Nexus9000 C9500v Chassis"
PID: N9K-C9500v , VID: , SN: 9NNVBPJ9N8A

NAME: "Slot 1", DESCR: "Nexus 9000v 64 port Ethernet Module"
PID: N9K-X9564v , VID: , SN: 9LKOT3OP6A5

NAME: "Slot 27", DESCR: "Supervisor Module"
PID: N9K-vSUP , VID: , SN: 9V3ZW14YP0M

switch#

```

## Cisco Nexus 9500v ラインカード

Cisco Nexus 9500v プラットフォームは、最大 16 枚の仮想ラインカードをサポートできます。このプラットフォームは、5つの異なる形式のラインカードをサポートしています。これらの仮想ラインカードの違いは、サポートする NX-OS インターフェイスの数です。デフォルトでは、Nexus 9500v プラットフォームは単一のラインカードで起動します。

CLI コマンドを使用して、仮想ラインカードを挿入または削除できます。シーケンシャル **vNIC マッピングスキーム**を使用する場合は、ラインカードをモジュール1から順番に挿入します。削除操作は逆の順序で行う必要があります。ただし、**MAC でエンコードされた vNIC マッピングスキーム**を使用する場合は、ラインカードを任意の順序で挿入できます。このモードは、ラインカードのまばらな実装をサポートします。

ラインカードのブートプロセスは、スーパーバイザが正常に起動して「アクティブ」状態に達した後に開始されます。対応するハードウェアと同様に、ラインカードの起動は「現在の」状態から開始され、「正常」状態に達すると完全に機能します。

ラインの挿入をサポートするには、**platform insert module number linecard** コマンドを使用します。**no** コマンドの前に **no** を付けると、ラインカードを削除できます。シャーシにラインカードを構成すると、シャーシの設定はスイッチの再起動後も維持されます。

```

switch# platform insert ?
      module  Insert a specific module

switch# platform insert module?
  <1-30>  Please enter the module number

switch# platform insert module 2?
  <CR>
      linecard  Linecard module

switch# platform insert module 2 linecard ?
      N9K-X9516v  Nexus 9000v 16 port Ethernet Module
      N9K-X9532v  Nexus 9000v 32 port Ethernet Module
      N9K-X9536v  Nexus 9000v 36 port Ethernet Module
      N9K-X9548v  Nexus 9000v 48 port Ethernet Module
      N9K-X9564v  Nexus 9000v 64 port Ethernet Module

```

## vNIC マッピング

実際の Cisco Nexus 9500 ハードウェアプラットフォームでは、ラインカードの前面パネルポートにファイバーを「プラグイン」できます。Nexus 9500v などの仮想プラットフォームでは、必要な数の仮想ネットワーク インターフェイス カード/インターフェイス (vNIC) をハイパーバイザーから Nexus 9500v プラットフォームにエクスポートします。

Nexus 9500v プラットフォームは、2 つの vNIC マッピング スキームをサポートしています。指定されたスキームに応じて、システムはハイパーバイザによって渡された vNIC を適切な Nexus 9500v NX-OS インターフェイスにマッピングします。次のセクションでは、両方の vNIC マッピング スキームの機能について説明します。

### シーケンシャル vNIC マッピング スキーム

このスキームでは、ハイパーバイザから取得した vNIC が NX-OS インターフェイスに順番にマッピングされます。たとえば、2 つの vNIC を Nexus 9500v にエクスポートすると、最初の vNIC が NX-OS 「mgmt」 インターフェイスにマッピングされます。2 番目の vNIC は、「Ethernet1/1」 インターフェイスにマップされます。これは、仮想スイッチが初めて起動するときのデフォルトのマッピング モードです。

その他の制限は次のとおりです。

- このモードは、ラインカードまたはインターフェイスのまばらな実装をサポートしていません。
- ラインカードは、モジュール 1 から順番に挿入し、逆の順序で取り外す必要があります。

ラインカードが取り外される場合、vNIC は引き続きシステム内に残り、ラインカードを再挿入すると、適切な NX-OS インターフェイスに自動的に再マッピングされます。

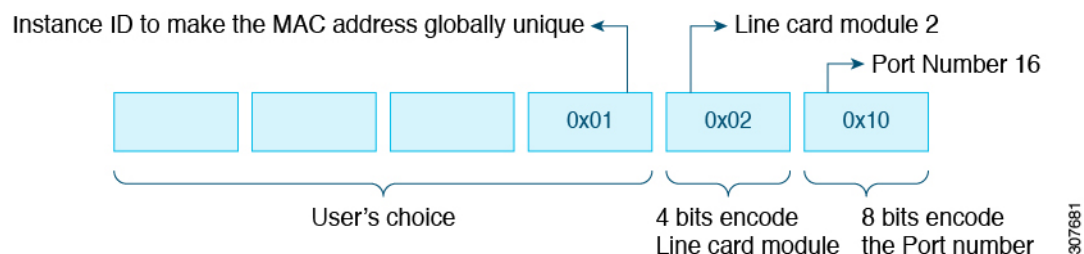
**platform vnic scheme** コマンドを使用して、vNIC マッピング スキームを選択します。マッピング スキームを選択すると、スイッチのリロードまで持続します。デフォルトで、Nexus 9500v スイッチはシーケンシャル vNIC マッピング スキームでぶーとアップします。このスキームの変更では、スイッチのリロードが必要です。

```
switch# platform vnic ?
  scheme      Virtual Network Interface Card allocation scheme

switch# platform vnic scheme ?
  mac-encoded  MAC address encoded allocation of VNICs to linecard modules
  sequential   Sequential allocation of VNICs to linecard modules
```

### MAC エンコード vNIC マッピング スキーム

このスキームでは、ハイパーバイザーから取得された vNIC は、vNIC に (ハイパーバイザ レベルで) 構成された MAC アドレスに基づいて NX-OS インターフェイスにマッピングされます。このモードでは、任意の vNIC を任意の NX-OS ライン カード インターフェイスにマッピングできます。このモードを使用するには、ライン カード モジュールとポート番号を vNIC MAC アドレスの最後の 2 バイトに追加します。この MAC アドレス構成は、Nexus 9500v 仮想スイッチの電源を入れる前にハイパーバイザ レベルで実行する必要があります。次の図は、必要な vNIC MAC アドレスの形式を示しています。



その他の機能：

- このモードは、ラインカードとインターフェイスのわずかな取り込みをサポートします。
- Nexus 9500v がこのモードになっている間は、ラインカードを任意の順序で挿入および取り外しできます。
- ラインカードが取り外される場合、vNIC は引き続きシステム内に残り、ラインカードを再挿入すると、適切な NX-OS インターフェイスに自動的に再マッピングされます。

**platform vnic scheme** コマンドを使用して、vNIC マッピング スキームを選択します。マッピング スキームを選択すると、スイッチのリロードまで持続します。デフォルトで、Nexus 9500v スイッチはシーケンシャル vNIC マッピング スキームでブートアップします。このスキームの変更では、スイッチのリロードが必要です。

```
switch# platform vnic ?
  scheme  Virtual Network Interface Card allocation scheme

switch# platform vnic scheme ?
  mac-encoded  MAC address encoded allocation of VNICs to linecard modules
  sequential    Sequential allocation of VNICs to linecard modules
```



(注) この方式では、ラインカードモジュールを任意の順序で挿入および削除できます。ただし、ラインカードを挿入した後は、どうしても必要な場合を除き、マッピング スキームをシーケンシャルに変更しないでください。マッピング スキームが「シーケンシャル」に変更された場合は、すべてのラインカードを順不同で取り外し、モジュール1から始めて挿入し直す必要があります。ラインカードを順不同で挿入し、スキームをシーケンシャルに変更してスイッチをリブートすると、どのラインカードもオンラインになりません。シーケンシャル スキームで起動されたスイッチは、既存のラインカードがモジュール1から始まるシリアル順に存在することを想定しています。次のようなエラー メッセージには、vNIC スキームと挿入されたラインカードの不一致が表示されます。

```
2020 Jul 15 14:44:03 N9Kv_3 %$ VDC-1 %$ %PLATFORM-2-MOD_INSERTION_FAILED:
Failed to insert module 6 (Nexus 9000v 64 port Ethernet Module - VNIC allocation scheme
is set to sequential, modules must be inserted in sequence)
```

**show platform vnic info** コマンドを入力して、スキームを検証できます。上記の状態から回復するには、**platform vnic scheme mac-encoded** コマンドを入力して vNIC スキームを MAC エンコードに戻し、スイッチを再起動します。vNIC スキームをシーケンシャルにする必要がある場合は、スキームをシーケンシャルに変更する前に、最初にすべてのラインカードを取り外します。

## vNIC Mapping Informational Show コマンド

## Show Platform vNIC コマンド

Cisco Nexus Nexus 9500v プラットフォームでは、CLI コマンドを使用して、現在の vNIC マッピングスキーム、マッピングされている vNIC の数、および MAC アドレスから vNIC へのマッピングを表示できます。これらのコマンドを使用すると、正しい数の vNIC が仮想マシンに渡されたことを確認でき、どのインターフェースがマップされているかを確認できます。

show vNIC platform コマンドの出力例:

## show platform vnic mapped

```
v-switch# show platform vnic mapped
NXOS Interface      VNIC MAC-Address      Internal VNIC
-----
Ethernet1/1         00c0.c000.0101        phyEth1-1
Ethernet1/2         00c0.c000.0102        phyEth1-2
Ethernet1/3         00c0.c000.0103        phyEth1-3
Ethernet1/4         00c0.c000.0104        phyEth1-4
Ethernet1/5         00c0.c000.0105        phyEth1-5
Ethernet1/6         00c0.c000.0106        phyEth1-6
Ethernet1/7         00c0.c000.0107        phyEth1-7
Ethernet1/8         00c0.c000.0108        phyEth1-8
Ethernet1/9         00c0.c000.0109        phyEth1-9
Ethernet1/10        00c0.c000.010a        phyEth1-10
Ethernet1/11        00c0.c000.010b        phyEth1-11
Ethernet1/12        00c0.c000.010c        phyEth1-12
Ethernet1/13        00c0.c000.010d        phyEth1-13
Ethernet1/14        00c0.c000.010e        phyEth1-14
Ethernet1/15        00c0.c000.010f        phyEth1-15
Ethernet1/16        00c0.c000.0110        phyEth1-16
```

## show platform vnic info

```
v-switch# show platform vnic info
VNIC Scheme: Mac-Encoded
mgmt0 interface: eth1 (00c0.c000.aabb)
Module      # VNICs Mapped
-----
1           16
-----
VNICs passed: 32
VNICs mapped: 16
VNICs unmapped: 16
```

## Nexus 9000v のスループット

このセクションでは、Nexus 9000v 仮想プラットフォームのおおよそのスループット値と、示された値を達成するためのリソース要件について説明します。

## 4vCPU/16GB

次の表は、4 つの仮想 CPU と 16 GB の RAM を備えた Nexus 9000v のおおよそのスループット値を示しています。

機能	ステートレス トラフィックのスループット	ステートフル トラフィックのスループット	
	サードパーティの <b>Packet-generator-tool</b> による <b>UDP</b> プロトコル トラフィック	ファイル転送による <b>TCP</b> プロトコル トラフィック	<b>iPerf-tool</b> による <b>TCP</b> プロトコル トラフィック
L2 スイッチング、ユニキャスト転送	~700 Mbps	~ 50 Mbps	500 Mbps
L3 ルーティング、ユニキャスト転送	60 Mbps	~ 5 Mbps	50 Mbps

スループット測定の詳細は次のとおりです：

トラフィック生成とスループットの測定には、iPerf ツールを備えた 2 つの Linux VM と 1 つのサードパーティ製デバイスを使用しました。

#### 使用されたトポロジ

- Nexus 9000v は、サードパーティのパケットジェネレータデバイスにバックツーバックで接続された、ESXI 8.0 ハイパーバイザ対応の UCS デバイスで実行されます。
- Nexus-9000v は Linux-VM にバックツーバックで接続され、すべての VM は ESXI 8.0 ハイパーバイザが有効な UCS デバイスで実行されました。

#### ステートレス（単方向）トラフィック テスト

サードパーティ デバイスによって生成されたユーザー定義の UDP パケットは、Nexus-9000v を介して送信されていました。

#### ステートフル（双方向）トラフィック テスト

- FTP ベースのファイル（2GB サイズ）転送は、Nexus-9000v を介してサードパーティ デバイスによって実行されました。
- iPerf-tools サーバー/クライアント VM によって生成された TCP トラフィックは、Nexus-9000v を介して送信されました。

## Nexus 9000v 機能のサポート

Cisco Nexus 9000v プラットフォーム ファミリは、幅広い Nexus 機能をシミュレートします。これらの機能の転送プレーンは、シスコ独自のソフトウェア データ プレーンに実装されています。したがって、動作が多少異なる場合があります。たとえば、システムスループットの量は、仮想シミュレーションプラットフォームとそれに対応するハードウェア間で異なります。

下表は Nexus 9000v プラットフォームでテストされた Cisco NX-OS の機能を示します。Nexus 9000v プラットフォームで、テストされていない NX-OS プラットフォームに依存しない機能を設定してシミュレートできます。ただし、これらの機能はサポートされていないと考えてください。プラットフォームでさらに多くの機能がテストされると、次の表が更新されます。

シャーシフォーム ファクタに依存する機能コマンドの一部は、対応する Nexus 9000v プラットフォームでのみ使用できることに注意することが重要です。たとえば、NAT コマンドは N9300 ハードウェアプラットフォームでのみ有効になり、N9500 ハードウェアプラットフォームでは有効になりません。コマンドが使用可能であるからといって、その機能がデータプレーンでサポートされるわけではないことに注意することも重要です。サポートされている機能については、次の機能表を参照してください。

## レイヤ2の機能

次の表に、Nexus 9300v および Nexus 9500v プラットフォームのレイヤ2機能のサポートを示します。

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
802.1AB LLDP	○	○
802.1Q VLAN/トランク	○	○
802.1s RST	○	○
802.3ad LACP	○	○
L2 マルチキャスト	はい（ブロードキャストとして）	はい（ブロードキャストとして）
vPC/MLAG	○	○
ポート チャンネル	○	○
VLAN	○	はい

## レイヤ3の機能

次の表に、Nexus 9300v および Nexus 9500v プラットフォームのレイヤ3機能のサポートを示します。

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
OSPF	○	○
OSPFv3	○	○
BGP	○	○

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
MP-BGP	○	○
IS-IS	はい（ブロードキャストとして）	はい（ブロードキャストとして）
RIPv2	○	○
等価コスト マルチパス ルーティング（ECMP）	○	○
PIM-SM	○	○
HSRP	○	○
VRRP	○	○
MPLS	○	○
EIGRP	○	○
CDP	○	○
L3 SVI	○	○
サブインターフェイス	○	○
IPSec	○	はい

## VXLAN およびセグメント ルーティング機能

次の表に、Nexus 9300v および Nexus 9500v プラットフォームの VXLAN およびセグメント ルーティング機能のサポートを示します。

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
VXLAN フラッドと BUM レプリケーションの学習 (PIM/ASM、IR)	(はい、はい)	(はい、はい)
VXLAN BGP eVPN レプリケーション (PIM/ASM、BIDIR、IR)	(はい、いいえ、はい)	(はい、いいえ、はい)
VXLAN EVPN ルーティング	○	○
VXLAN EVPN ブリッジング	○	○

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
VXLAN EVPN エニーキャスト GW	はい（ブロードキャストとして）	はい（ブロードキャストとして）
VXLAN トンネル エンドポイント	○	○
VXLAN ARP を抑制	○	○
VXLAN EVPN マルチサイト BGW	はい (ボーダー ゲートウェイ (BGW) の非 vPC を使用)	はい (ボーダー ゲートウェイ (BGW) の非 vPC を使用)
VXLAN EVPN TRM	非対応	非対応
VXLAN IPv6 アンダーレイ	○	○
MPLS セグメント ルーティング (SRv4)	非対応	非対応
ダウンストリーム VNI	○	○
ファブリック ピアリングが設定された vPC <a href="#">1</a>	○	○
L3VNI	非対応	非対応

<sup>1</sup> vPC ファブリック ピアリング ピアリンクは、トランスポートネットワーク（ファブリックのスパイン層）を介して確立されます。DSCP は N9000v/N9300v/N9500v ではサポートされていません。したがって、スパインがトラフィックバーストで過負荷になると、リーフスイッチの vPC 状態が中断する可能性があります。

## プログラマビリティ機能

次の表に、Nexus 9300v および Nexus 9500v プラットフォームのプログラマビリティ機能のサポートを示します。

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
Bash シェルアクセスとスク립ティング	○	○
RPM サポート	○	○
システム状態へのプログラムによるアクセス (Python)	○	○

機能	Nexus 9300v のサポート	Nexus 9500v のサポート
OS 内のゲスト シェル	○	○
OS 内の Docker	○	○
NXAPI	○	○
DME	○	○
RESTCONF	○	○
NETCONF	○	○
YANG モデル	○	○
テレメトリ	○	○
GNMI	○	○
NxSDK	○	はい

## Guestshell のサポート

Cisco Nexus 9000v は Nexus Guestshell をサポートしています。しかし、この機能はデフォルトで有効になっています。有効にするために適切な Nexus Guestshell ドキュメントに従って、Guestshell 機能を使用できます。



- (注) 現在、Cisco Nexus 9000v 仮想アーティファクトには、2つのバイナリ イメージを収めるのに十分なブートフラッシュ サイズがあります。ただし、Guestshell を有効にすると、より多くのブートフラッシュ ディスク サイズが必要になります。リリースされた仮想アーティファクトの2つのバイナリ イメージをそれぞれブートするための十分な領域がブートフラッシュにない可能性があります。Guestshell を有効にする必要がある場合は、十分なディスク領域を空けるように計画してください。

## Nexus 9000v Platform MIB のサポート

Nexus 9000v プラットフォームは、多くの Cisco NX-OS SNMP MIB とともに Simple Network Management Protocol (SNMP) をサポートしています。一部の管理対象オブジェクトは、シミュレーションプラットフォームに関連していない可能性があり、サポートされていない可能性があることに注意してください。

次のリストは、サポートされているプラットフォーム関連の MIB を示しています。

- CISCO エンティティ アセット MIB

- ceEXTEntityLEDTable
- ciscoEntityExtMIB
- ciscoRFMIB
- ciscoTSMIB
- ciscoEntityFRUControlMIB
- ciscoSyslogMIB

## Nexus 9000v プラットフォームの注意事項と制限事項

次のガイドラインおよび制約事項が Nexus 9000v に適用されます。

- ハードウェア整合性チェッカーはサポートされていません
- 無停止 ISSU はサポートされていません
- NX-OS インターフェイスのリンク キャリア ステータスは、基礎となるハイパーバイザの機能に依存します。

ネストされた VM のユース ケースがサポートされています。ただし、要件に基づいて、さまざまな環境に Nexus 9000v VM をデプロイすることに注意してください。ネストされた VM 環境では、使用可能な vCPU とメモリ リソースに応じて、起動時間、管理操作、および機能アプリケーションのパフォーマンスが低下します。

- 多数のノード、または 10 を超える Nexus 9000v ノードをシミュレートする場合は、ベアメタル環境を使用します。VIRL2 は現在サポートされていません。
- 多数のノードをシミュレートすると、起動時間が長くなります。
- スパース モードは現在、VIRL 環境ではサポートされていません。
- Cisco NX-OS リリース 10.1(1) 以降、ESXi で同時に起動するために複数の VM が必要な場合は、一度に 2～3 台の VM のクラスタを起動することが推奨され、一定の時間間隔を置いてから、次の 2～3 台の VM のクラスタを起動する必要があります。そうしないと、VM が応答しなくなる可能性があります。
- Cisco NX-OS リリース 10.2(3)F 以降、fcoe/fcoe-npv フィーチャ セットのサポートが Cisco Nexus 9300v プラットフォーム スイッチに追加されましたが、9500v プラットフォーム スイッチには追加されません。
- Cisco NX-OS リリース 10.2 (3) F 以降、メモリ フットプリントを削減した Lite イメージが導入され、Cisco Nexus 9300v および 9500v の以前のイメージよりもはるかに小さいイメージのサイズになります。
- テレメトリは N9kv lite イメージではサポートされていないため、gNMI テストは N9kv lite イメージでは実行できません。しかし、Cisco NX-OS リリース 10.4 (2) F 以降では、これは、サポートされています。

- Cisco NX-OS リリース 10.6(1)F 以降、MLD は Cisco N9000V スイッチでサポートされます。詳細については、『[Cisco Nexus 9000 シリーズ NX-OS マルチキャスト ルーティング構成ガイド](#)』の「[MLD の構成](#)」を参照してください。
- ユーザーが ESXi バージョンを 8.0 以降にアップグレードする場合は、ESXi ホスト管理 UI の [管理 (Manage)] > [システム (System)] > [詳細 (Advance)] 設定で Net.BlockGuestBDPU の既存の値 (1) をゼロ (0) に変更して、LACP が N9kv で機能するようにします。



## 第 3 章

# Nexus 9000v の展開

この章は、次の項で構成されています。

- [Nexus 9000v ハイパーバイザ サポート \(23 ページ\)](#)
- [KVM/QEMU に対する Nexus 9000v 展開ワークフロー \(25 ページ\)](#)
- [ESXi の Nexus 9000v 展開ワークフロー \(31 ページ\)](#)
- [Vagrant 用 Nexus 9000v 展開ワークフロー \(33 ページ\)](#)
- [イメージアップグレードのワークフロー \(38 ページ\)](#)

## Nexus 9000v ハイパーバイザ サポート

Nexus 9000v プラットフォーム ファミリの両方のプラットフォームは、サポートされているハイパーバイザで仮想マシンとして実行するように設計されています。基盤となるハイパーバイザの制限により、プラットフォーム機能の一部が制限される場合があります。このセクションでは、サポートのレベルと関連する制限について説明します。

## KVM/QEMU 属性

次の表に、KVM/QEMU ハイパーバイザでサポートされる属性を示します。

属性	サポート
QEMU バージョン	4.2.0

属性	サポート
BIOS	<p>OVMF バージョン 16、  <a href="https://www.kraxel.org/repos/jenkins/edk2/">https://www.kraxel.org/repos/jenkins/edk2/</a></p> <p>この URL は、最新の OVMF RPM パッケージ ファイルを含むインデックス ページにアクセスします。ファイルの例は次のとおりです。</p> <p><code>edk2.git-ovmf-x64-0-20200515.1388.g9099dcbd61.noarch.rpm</code></p> <p>RPM ユーティリティを使用してパッケージ ファイルをダウンロードして抽出します。パッケージにはいくつかのファイルが含まれています。OVMF-pure-efi.fd を見つけて、BIOS ファイルとして使用します。必要に応じて、名前を bios.bin に変更できます。</p>
Linux バージョン	Ubuntu 20.0.4
プラットフォーム (Platform)	<p>Nexus 9300v の展開</p> <p>Nexus 9500v の展開</p>
ラインカード	<p>Nexus 9300v: 1 枚のラインカード</p> <p>Nexus 9500v: 最高 16 枚のラインカード</p>
ライン カード インターフェイス	<p>Nexus 9300v: 最大 64 枚のライン カード インターフェイス</p> <p>Nexus 9500v: 最大 400 枚のライン カード インターフェイス</p>

## ESXI 属性

次の表に、ESXI ハイパーバイザでサポートされる属性を示します。

属性	サポート
version	8.0。
プラットフォーム (Platform)	<p>Nexus 9300v の展開</p> <p>Nexus 9500v の展開</p>
ライン カード	<p>Nexus 9300v: 1 枚のラインカード</p> <p>Nexus 9500v: 最高 16 枚のラインカード</p>

属性	サポート
ラインカード インターフェイス	Nexus 9300v: 最大9つのラインカードインターフェイス  Nexus 9500v: 最大9つのラインカードインターフェイス

## VirtualBox 属性

次の表に、VirtualBox ハイパーバイザでサポートされている属性を示します。

属性	サポート
version	7.0
プラットフォーム (Platform)	Nexus 9300v の展開
ライン カード	Nexus 9300v: 1 枚のラインカード
ラインカード インターフェイス	Nexus 9300v: 最大 4 個のライン カード インターフェイス

## KVM/QEMU に対する Nexus 9000v 展開ワークフロー

このセクションでは、KVM/QEMU ハイパーバイザに Nexus 9000v プラットフォームを展開するために必要な手順について説明します。次の 3 種類の展開を使用できます。

- 共通展開
- プラットフォーム特有の展開
- インターコネクトの展開

## 共通展開ワークフロー

KVM/QEMU ハイパーバイザを介して Cisco Nexus 9000v プラットフォームを展開できます。次の表に、KVM/QEMU での Cisco Nexus 9000v 展開でサポートされるパラメータを示します。

パラメータ	例	説明
/path_to/qemu	/usr/bin/qemu-system-x86_64	QEMU の実行可能ファイルへのパス。(さまざまなバージョンの QEMU ソフトウェアを <a href="http://wiki.qemu.org/download">http://wiki.qemu.org/download</a> からダウンロードします。)

パラメータ	例	説明
-nographic	-nographic	Cisco Nexus 9000v プラットフォームは VGA をサポートしていないため、推奨されません。
-bios file	-bios bios.bin	<p>必須。Cisco Nexus 9000v プラットフォームは EFI ブートを使用し、動作するには互換性のある BIOS イメージが必要です。</p> <p>ディスク操作のパフォーマンスを向上させるには、SATA コントローラで最新の OVMF BIOS ファイルを使用することをお勧めします。SATA コントローラでは QEMU 2.6 を推奨します。詳細については、<a href="http://www.linux-kvm.org/page/OVMF">http://www.linux-kvm.org/page/OVMF</a> を参照してください。</p>
-smp	-smp 4	Cisco Nexus 9000v プラットフォームは、1 ～ 4 個の vCPU をサポートします（2 ～ 4 個をお勧めします）。
-m memory	-m 10240	メモリ（MB）。
-serial telnet:host:port,server,nowait	-serial telnet:localhost:8888,server,nowait  または  -serial telnet:server_ip:8888,server,nowait	少なくとも 1 つを指定します。

パラメータ	例	説明
-net ... -net ... または -netdev ... -device ...	<pre> -net socket,vlan=x,name=nl_s0,listen= localhost:12000  -net nic, vlan=x, model=e1000, macaddr=aaaa.bbbb.cccc  -netdev socket,listen=localhost:12000,id=eth_s_f  -device e1000,addr=s.f,netdev=eth_s_f, mac=aaaa.bbbb.cccc, multifunction=on,romfile=  または  -netdev tap,ifname=tap_s_f,script=no, downscript=no,id=eth_s_f  -device e1000,addr=s.f,netdev=eth_s_f, mac=aaaa.bbbb.ccc, multifunction=on,romfile= </pre>	<p>net/net または netdev/device のペアは、仮想ネットワーク インターフェイス カード (vNIC) をネットワーク化するためのものです。</p> <p>_s_f は、PCI スロット番号と機能番号を表します。QEMU 2.0 以降では、少なくとも 20 個の PCI スロットと 4 つの機能をプラグインでき、合計で約 80 個の vNIC に対応できます。スロットの範囲は 3 ~ 19、関数番号の範囲は 0 ~ 3 です。</p> <p>mac= オプションは、各 vNIC MAC アドレスの MAC アドレスを VM インターフェイスに渡します。最初の -netdev は、VM の mgmt0 インターフェイスに自動的にマップされます。2 番目の -netdev は e1/1 インターフェイスにマップされ、e1/64 の 65 番目まで同様にマップされます。MAC アドレスがネットワーク デバイスごとに一意であることを確認します。</p>
-enable-kvm	-enable-kvm	このフラグは、Cisco Nexus 9000v に必要です。
-drive ... -device ... (SATA コントローラの場合)	<pre> -device ahci, id=ahci0,bus=pci.0  -drive file=img.qcow2, if=none,id=drive-sata-disk0, format=qcow2  -device ide-hd,bus=ahci0.0, drive=drive-sata-disk0, id=drive-sata-disk0 </pre>	<p>SATA コントローラに使用するフォーマット。QEMU 2.6.0 で SATA コントローラを使用することをお勧めします。これは、このコントローラが IDE コントローラよりも優れたパフォーマンスを提供するためです。ただし、SATA コントローラをサポートしていない QEMU の早期バージョンがある場合は、IDE コントローラを使用できます。</p>

パラメータ	例	説明
-drive ... media=cdrom	-drive file=config.iso,media=cdrom	<p>Cisco Nexus 9000v プラットフォームの起動後に適用されるスイッチ構成ファイルを含む CD-ROM ディスク。</p> <p>1. テキスト ファイルに名前を付けます (nxos_config.txt)。</p> <p>2. Linux の mkisofs -o config.iso -l --iso-level 2 nxos_config.txt コマンドを使用して、config.iso を作成します。</p>

## プラットフォーム特有のワークフロー

Cisco Nexus 9500v プラットフォームは、シーケンシャル モードと MAC エンコード モードの 2 つの異なるモードで実行されます。Nexus 9300v および Nexus 9500v のシーケンシャル モードの展開手順は、KVM/QEMU ハイパーバイザでまったく同じです。この場合、両方のプラットフォームの最大インターフェイスは 401 インターフェイスです (1 つの管理ポートまたは 400 のデータ ポート)。

Nexus 9500v は、複数のラインカードでインターフェイス トラフィックをエミュレートします。仮想スイッチは、最大合計 400 のインターフェイスに対して KVM/QEMU 上の単一の VM を使用します。Nexus 9500v の MAC エンコード スキーマに基づいて、KVM/QEMU CLI コマンドが呼び出されたときに、エンコードされたスロットとポート番号を使用して各ネットワーク アダプターの MAC アドレスを指定します。

## プラットフォームのインターコネクト

Nexus 9000v プラットフォーム インスタンスまたはその他の仮想プラットフォーム間のインターコネクトは、Linux ブリッジとタップに基づいています。CLI コマンドを呼び出す前に、以下が利用可能であることを確認してください (構成例が提供されています)。

以下の構成例では、ブリッジとタップインターフェイスを、それぞれ 1 つの管理インターフェイスと 1 つのデータ インターフェイスを持つ 2 つの N9Kv スイッチとともに作成できます。管理インターフェイス「interface mgmt0」は、ブリッジ「mgmt\_bridge」を使用して管理ネットワークに接続されています。両方のスイッチからのデータ ポート インターフェイス「interface Eth1/1」は、ブリッジ「interconnect\_br」を使用して背中合わせに接続されます。



(注) 必要な最小 QEMU バージョンは、Cisco NX-OS リリース 9.3(3) 以降の 4.2.0 です。

- ブリッジ (ESXi ハイパーバイザーの vSwitch に類似) が作成され、「UP」状態に設定されます。

ブリッジを作成して UP 状態にする Linux コマンド:

```
sudo brctl addbr mgmt_bridge
sudo brctl addbr interconnect_br
sudo ifconfig mgmt_bridge up
sudo ifconfig interconnect_br up
```

- タップ インターフェイスは、Nexus 9000v が使用しているインターフェイスの数に基づいて作成されます。

タップ インターフェイスを作成する Linux コマンド:

```
sudo openvpn --mktun --dev tap_sw1_mgmt
sudo openvpn --mktun --dev tap_sw2_mgmt
sudo openvpn --mktun --dev tap_sw1_eth1_1
sudo openvpn --mktun --dev tap_sw2_eth1_1
```

- ブリッジはタップ インターフェイスに接続されます。

ブリッジをタップ インターフェイスに接続する Linux コマンド:

```
sudo brctl addif mgmt_bridge tap_sw1_mgmt
sudo brctl addif mgmt_bridge tap_sw2_mgmt
sudo brctl addif interconnect_br tap_sw1_eth1_1
sudo brctl addif interconnect_br tap_sw2_eth1_1
```

- すべてのタップ インターフェイスは「UP」状態である必要があります。

タップ インターフェイスを UP 状態にする Linux コマンド:

```
sudo ifconfig tap_sw1_mgmt up
sudo ifconfig tap_sw2_mgmt up
sudo ifconfig tap_sw1_eth1_1 up
sudo ifconfig tap_sw2_eth1_1 up
```

- すべてのタップ インターフェイスがブリッジに接続されていることを確認します

タップ インターフェイスがブリッジに接続されていることを確認する Linux コマンド:

```
brctl show
```

bridge name	bridge id	STP enabled	interfaces
interconnect_br	8000.1ade2e11ec42	no	tap_sw1_eth1_1 tap_sw2_eth1_1
mgmt_bridge	8000.0a52a9089354	no	tap_sw1_mgmt tap_sw2_mgmt

2 つの Nexus 9000v プラットフォームを起動し、それぞれ 1 つのインターフェイスを背中合わせに接続するには、次のコマンドを例として使用できます。接続は、ソケットベースまたはブリッジベースの接続にすることができます。この例では、ブリッジを使用して、管理インター

フェイスのインスタンスと 1 つのデータ ポートを接続します。同様に、コマンドライン オプションでネット デバイスを追加することで、同じ方法でより多くの Nexus 9000v データ ポートを接続できます。この例では、両方の Nexus 9000v インスタンスの 2 つのインターフェイス (インターフェイス mgmt0 とインターフェイス eth1/1) がそれぞれマッピングされています。

Nexus 9000v の最初のインスタンスの場合:

```
sudo qemu-system-x86_64 -smp 2 -m 8196 -enable-kvm -bios bios.bin
-device i82801b11-bridge,id=dmi-pci-bridge
-device pci-bridge,id=bridge-1,chassis_nr=1,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-2,chassis_nr=2,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-3,chassis_nr=3,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-4,chassis_nr=4,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-5,chassis_nr=5,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-6,chassis_nr=6,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-7,chassis_nr=7,bus=dmi-pci-bridge
-netdev tap,ifname=tap_sw1_mgmt,script=no,downscript=no,id=eth1_1_0
-device e1000,bus=bridge-1,addr=1.0,netdev=eth1_1_0,mac=00:b0:b0:01:aa:bb,multifunction=on,romfile=
-netdev tap,ifname=tap_sw1_eth1_1,script=no,downscript=no,id=eth1_1_1
-device e1000,bus=bridge-1,addr=1.1,netdev=eth1_1_1,mac=00:b0:b0:01:01:01,multifunction=on,romfile=
-device ahci,id=ahci0 -drive
file=test1.qcow2,if=none,id=drive-sata-disk0,id=drive-sata-disk0,format=qcow2
-device ide-hd,bus=ahci0.0,drive=drive-sata-disk0,id=drive-sata-disk0
-serial telnet:localhost:9000,server,nowait -M q35 -daemonize
```

Nexus 9000v の 2 番目のインスタンスの場合:

```
sudo qemu-system-x86_64 -smp 2 -m 8196 -enable-kvm -bios bios.bin
-device i82801b11-bridge,id=dmi-pci-bridge
-device pci-bridge,id=bridge-1,chassis_nr=1,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-2,chassis_nr=2,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-3,chassis_nr=3,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-4,chassis_nr=4,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-5,chassis_nr=5,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-6,chassis_nr=6,bus=dmi-pci-bridge
-device pci-bridge,id=bridge-7,chassis_nr=7,bus=dmi-pci-bridge
-netdev tap,ifname=tap_sw2_mgmt,script=no,downscript=no,id=eth1_1_0
-device e1000,bus=bridge-1,addr=1.0,netdev=eth1_1_0,mac=00:b0:b0:02:aa:bb,multifunction=on,romfile=
-netdev tap,ifname=tap_sw2_eth1_1,script=no,downscript=no,id=eth1_1_1
-device e1000,bus=bridge-1,addr=1.1,netdev=eth1_1_1,mac=00:b0:b0:02:01:01,multifunction=on,romfile=
-device ahci,id=ahci0 -drive
file=test2.qcow2,if=none,id=drive-sata-disk0,id=drive-sata-disk0,format=qcow2
-device ide-hd,bus=ahci0.0,drive=drive-sata-disk0,id=drive-sata-disk0
-serial telnet:localhost:9100,server,nowait -M q35 -daemonize
```

qemu-system-x86\_64 以上の KVM コマンドは、Linux のデプロイ方法によっては同等です。呼び出しが成功すると、「telnet localhost 9000」または「telnet localhost 9100」を介して、シリアル コンソールの両方のインスタンスにそれぞれアクセスできるはずです。

LLDP および LACP マルチキャスト固有のパケットのトラフィックを Linux ブリッジ経由で渡すには、各インスタンスに接続するすべてのブリッジで次の値を設定します。

- VM 間の LLDP および LACP 通信を設定します。

```
echo 0x4004 > /sys/class/net/br_test/bridge/group_fwd_mask
```

- Linux ブリッジを介したマルチキャスト パケット フローを許可します。

```
echo 0 > /sys/devices/virtual/net/br_test/bridge/multicast_snooping
```

## ESXi の Nexus 9000v 展開ワークフロー

このセクションでは、ESXi ハイパーバイザに Nexus 9000v プラットフォームを展開するために必要な手順について説明します。次の 3 種類の展開を使用できます。

- 共通展開
- プラットフォーム特有の展開
- インターコネクトの展開

### 共通展開ワークフロー

#### 始める前に

次の手順では、分散 OVA を使用して、ESXi ハイパーバイザで Cisco Nexus 9300v または 9500v プラットフォームをプロビジョニングします。

次の状態を確認してください。

- ESXi 8.0 ハイパーバイザをインストールしている
- サーバーと vCenter の両方で実行する ESXi 8.0 の有効なライセンスがあります。
- 配布された OVA ファイルがデスクトップにダウンロードされていること。

#### 手順

- 
- ステップ 1** ESXi vCenter にログインします。
- ステップ 2** バージョン 8.0 を右クリックして **[OVF テンプレートの展開 (Deploy OVF Template)]** を選択します。
- (注)  
表示される後続の画面でセルフガイドの指示を実行します。
- ステップ 3** **[名前が必要 (Need name)]** 画面で、**[ローカル ファイル (Local file)]** を選択し、**[参照 (Browse)]** をクリックします。デスクトップからダウンロードした配布 OVA ファイルを選択します。
- ステップ 4** **[need name]** 画面で、データセンター (またはフォルダ) を選択し、VM 名を入力します。
- ステップ 5** **[名前が必要 (Need name)]** 画面で、仮想マシンを展開する ESXi サーバーを選択し、検証後に **[完了 (Finish)]** をクリックします。
- ステップ 6** **[名前が必要 (Need name)]** 画面で、詳細を確認し、**[次へ (Next)]** をクリックします。
- ステップ 7** **[Configure]** 画面で、**[次へ]** をクリックします。

**ステップ 8** [Select Storage] 画面で、データ ストアを選択し、[次へ]をクリックします。

**ステップ 9** [ネットワークの選択 (Select Networks)] 画面で、次の値が選択されていることを確認します。

- 送信元ネットワーク名 : mgmt 0
- 宛先ネットワーク : ラボ管理 LAN vSwitch

ラボ管理 LAN vSwitch として他の vNIC 宛先を選択しないでください。そうしないと、Cisco Nexus 9000v データ ポートが物理スイッチと競合するため、管理接続の問題が発生します。

**ステップ 10** [Ready to Complete] 画面で、[Finish] をクリックし、プロセスが完了するまで待ちます。

**ステップ 11** [仮想ハードウェア] タブで、[シリアル ポート 1] を選択します。シリアル ポート タイプについては、[ネットワークの使用] パネルを選択し、次のオプションを選択します。

- 方向 - サーバー
- ポート URL - telnet://0.0.0.0:1000。1000 はこのサーバーの一意のポート番号です。

(注)

Nexus 9000v は、E1000 ネットワーク アダプタのみをサポートします。ネットワーク アダプターを追加するときは、アダプターの種類が E1000 であることを確認します。

**ステップ 12** [VM Options] タブで、[Boot Options] パネルを選択し、[EFI] を選択します。

**ステップ 13** [VM Options] タブで、[Advance] パネルを選択し、[Edit Configuration] 画面で、[Add Configuration Params] オプションを使用して次の値を追加します。

- 名前 - efi.serialconsole.enabled
- 値 - TRUE

[OK] をクリックして、VGA とシリアル コンソール モードの両方で起動プロセスを表示します。

(注)

Nexus 9000v プラットフォームでは、スイッチ プロンプトにアクセスするためにシリアル コンソールをプロビジョニングする必要があります (ただし、最初の grub ブート メッセージの一部は VGA コンソールに表示されます)。シリアル コンソールが VM に正しくプロビジョニングされていることを確認します。「efi.serialconsole.enabled=TRUE」がプロビジョニングされている場合、VGA またはシリアル コンソールから「Nexus9000v のイメージ署名検証が実行されていません」が表示された後、ブートアップが成功するとカーネル ブート メッセージが表示されます。

**ステップ 14** 仮想マシンの電源をオンにします。

## プラットフォーム特有のワークフロー

Cisco Nexus 9500v は、シーケンシャル モードと MAC エンコード モードの 2 つの異なるモードで動作します。Nexus 9300v および Nexus 9500v のシーケンシャル モードの展開手順は、ESXi ハイパーバイザでまったく同じです。両方のプラットフォームタイプのインターフェイスの最

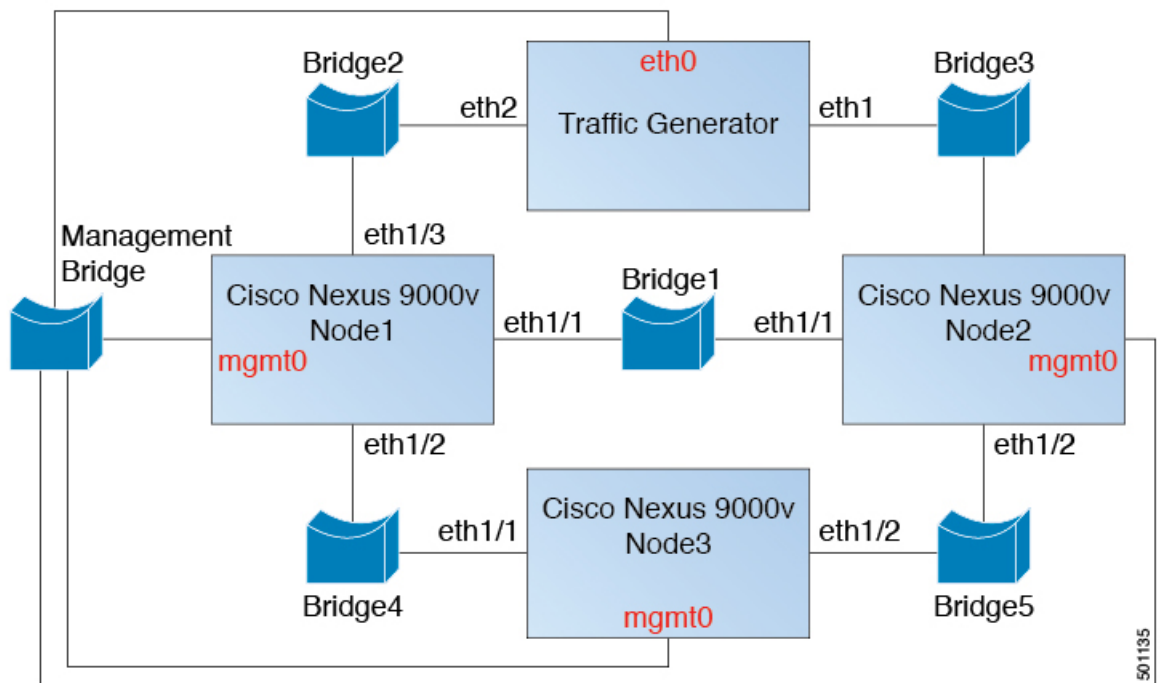
大数は 10（1 つの管理ポートと 9 つのデータ ポート）です。これはハイパーバイザの制限です。

Nexus 9500v は、インターフェイスの総数が 10 に制限されている場合でも、ESXi ハイパーバイザ上の単一の VM で複数回線カードインターフェイストラフィックをエミュレートします。Nexus 9500v の MAC エンコードスキーマの使用を選択した場合は、エミュレートされているスロットとポートに一致するように各ネットワークアダプタの MAC アドレスを変更します。

## プラットフォームのインターコネクト

Nexus 9300v と Nexus 9500v、またはその他の仮想プラットフォーム間のネットワークは、ESXi ハイパーバイザのブリッジとしての vSwitch に基づいています。さまざまな顧客のユースケースをシミュレートするように設計された任意のトポロジを使用できます。

図 1: ESXi を介した Cisco Nexus 9000v プラットフォームのインターコネクト



## Vagrant 用 Nexus 9000v 展開ワークフロー

このセクションでは、Vagrant ハイパーバイザーに Nexus 9000v プラットフォームを展開するために必要な手順について説明します。次の 3 種類の展開を使用できます。

- 共通展開
- プラットフォーム特有の展開
- インターコネクトの展開

## 共通展開ワークフロー

Vagrant/VBox 環境に Cisco Nexus 9300v を展開することはできません。仮想 artifacts.box ファイルは、配布でのみ入手できます。

## プラットフォーム特有のワークフロー

nexus9300v.9.3.3.IDI9.0.XXX.box を VirtualBox に展開します。Vagrant/Vbox の使用に関する次のカスタマイズ ガイドラインと警告を参照してください。

- Vagrant ファイルでのユーザーのカスタマイズは必要ありません。
- Windows の名前付きパイプを変更する必要はありません。Mac または Windows の両方で、デフォルトのポート 2023 を使用してシリアル コンソールにアクセスします。必要に応じて、このシリアル コンソールを **telnet localhost 2023** 経由で使用して、スイッチの起動プロセスを監視します。
- 標準のボックスプロセスは、他のアプライアンスのディストリビューションと同様に使用されます。ベース ボックス名を使用して VM を起動するだけです。
- ボックス名は、Vagrant ファイルの **config.vm.box** フィールドを使用して、「base」以外の別の名前に変更できます。
- ブートストラップ構成は、リリース イメージファイルから .box の既存の汎用構成以外に、スイッチに別の構成を適用する場合に可能です。この場合、**vb.customize pre-boot** を使用します。例：

```
vb.customize "pre-boot", [
    "storageattach", :id,
    "--storagectl", "SATA",
    "--port", "1",
    "--device", "0",
    "--type", "dvddrive",
    "--medium", "../common/nxosv_config.iso",
```

- **config.vm.base\_mac** フィールドを使用して、VM インターフェイスの MAC アドレスをカスタマイズします。この変更は、**vagrant up** CLI コマンドを入力する前、および **vagrant init** CLI コマンドを入力した後に実行する必要があります。**vagrant up** CLI コマンドの入力後、または VM の作成後に MAC アドレスを変更する場合は、ボックス コマンドを使用して VM を変更します。

## Vagrant での Sync フォルダのサポート

リリース 10.1(1) 以降、Nexus 9300v は、ホスト マシン上のディレクトリ/フォルダを Nexus 9300v マシンと共有できる Vagrant 同期フォルダをサポートします。Vagrant スクリプトの **vagrant up** コマンドは仮想ボックスにログインし、Vagrantfile のユーザー構成に基づいてディレクトリをマウントします。デフォルトでは、Vagrant スクリプトは **vagrant** ユーザー名を使用し、**bash** がログイン シェルであることを想定しています。この機能を容易にするために、事前設定された **vagrant** ユーザー名のデフォルトのログインシェルが **bash** に変更されました。

ただし、デフォルトのシェル (ユーザー *vagrant* の場合) を、Nexus または Vagrantfile で明示的に設定して NX-OS CLI に変更するオプションがあります。

デフォルトでは、Vagrant はホストの現在の作業ディレクトリを Nexus 9300v の *directory/vagrant* にマウントします。ホスト上の現在のフォルダを Nexus 9300v と共有したくない場合は、Vagrantfile に次の行を含める必要があります。

```
config.vm.synced_folder ".", "/vagrant", disabled: true
```

サンプル Vagrantfile - ホスト フォルダを共有する場合、たとえば、Nexus 9300v の */bootflash/home/vagrant* の */home/james/my\_shared\_folder/* :

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.

  config.vm.define "n9kv1" do |n9kv1|

    n9kv1.vm.box = "10.1.1"

    n9kv1.ssh.insert_key = false
    n9kv1.vm.boot_timeout = 600

    if Vagrant.has_plugin?("vagrant-vbguest")
      config.vbguest.auto_update = false
    end

    config.vm.synced_folder ".", "/vagrant", disabled: true
    config.vm.synced_folder "/home/james/my_shared_folder" "/bootflash/home/vagrant/"

    config.vm.box_check_update = false

  end
end
```

以下に、Nexus 9300v プラットフォーム固有の展開例を示します。

```
vagrant box add 10.1.1 nexus9300v.10.1.1.box

$ vagrant init 10.1.1
$ vagrant up

Bringing machine 'n9kv1' up with 'virtualbox' provider...
==> n9kv1: Importing base box '10.1.1'...
==> n9kv1: Matching MAC address for NAT networking...
==> n9kv1: Setting the name of the VM: vagrant_n9kv1_1605848223701_17342
==> n9kv1: Clearing any previously set network interfaces...
==> n9kv1: Preparing network interfaces based on configuration...
n9kv1: Adapter 1: nat
==> n9kv1: Forwarding ports...
n9kv1: 22 (guest) => 2222 (host) (adapter 1)
==> n9kv1: Booting VM...
==> n9kv1: Waiting for machine to boot. This may take a few minutes...
n9kv1: SSH address: 127.0.0.1:2222
n9kv1: SSH username: vagrant
n9kv1: SSH auth method: private key
```

```

==> n9kv1: Machine booted and ready!
==> n9kv1: Checking for guest additions in VM...
n9kv1: The guest additions on this VM do not match the installed version of
n9kv1: VirtualBox! In most cases this is fine, but in rare cases it can
n9kv1: prevent things such as shared folders from working properly. If you see
n9kv1: shared folder errors, please make sure the guest additions within the
n9kv1: virtual machine match the version of VirtualBox you have installed on
n9kv1: your host and reload your VM.
n9kv1:
n9kv1: Guest Additions Version: 5.2.18 r123745
n9kv1: VirtualBox Version: 6.1
==> n9kv1: Mounting shared folders...
n9kv1: /bootflash/home/vagrant => /home/james/my_shared_folder

$ vagrant ssh

-bash-4.4$

```

## デフォルトのシェルの NX-OS CLI への変更

NX-OS CLI にログインする必要がある場合は、次のいずれかのオプションを使用します。

- ログインするたびに **bash** プロンプトで **vsh** コマンドを手動で実行します。
- 以下に示すように、Nexus 9300v 仮想ボックスに事前にパッケージ化されたスクリプトを使用して、Vagrantfile から実行することができます。

```

config.vm.synced_folder ".", "/vagrant", disabled: true
config.vm.synced_folder "/home/james/my_shared_folder"
"/bootflash/home/vagrant/"
config.vm.box_check_update = false

config.vm.provision "shell", inline: "vsh -r
/var/tmp/set_vsh_as_default.cmd"

```

- ユーザー名 *vagrant* の代わりにユーザー名 *admin* でログインできます (**vagrant ssh** コマンドを使用すると、デフォルトでユーザー名 *vagrant* が使用されます)。

```
ssh -p 2222 admin@127.0.0.1
```

## Nexus 9300v での Ansible の使用

Vagrant は、Ansible、Shell スクリプト、Ruby スクリプト、Puppet、Chef、Docker、Salt などのさまざまなプロビジョナーを使用して、ボックスの構成と管理をサポートする汎用オーケストレーターです。

Vagrant ファイルには、1 つ (または複数) のプロビジョナーのセクションとその構成が含まれている場合があります。Ansible の例をここに示します。

```

n9kv1.vm.provision "ansible" do |ansible|
  ansible.playbook = "n9kv1.yml"
  ansible.compatiblity_mode = "2.0"
end

```

これらのプロビジョナーは、仮想ボックスが起動するたびに、または **vagrant provision** コマンドまたは **vagrant provision --provision-with** コマンドを使用して手動でトリガーされたときに、自動的にトリガーされます。Ansible が仮想ボックスにログインして NX-OS CLI を実行するに

は、Ansible ホスト構成ファイルでログイン資格情報を提供します。Ansible はログイン後に NX-OS CLI が表示されることを想定しているため、事前構成されたユーザー名 *admin* を使用するか、新しいユーザー名を手動で作成して、Ansible ホスト構成ファイルで使用できます。

### VM のシャットダウン

次を使用して VM をシャットダウンします。

```
$ vagrant halt -f
==> default: Forcing shutdown of VM...
```

### クリーンアップのために VM を破棄する

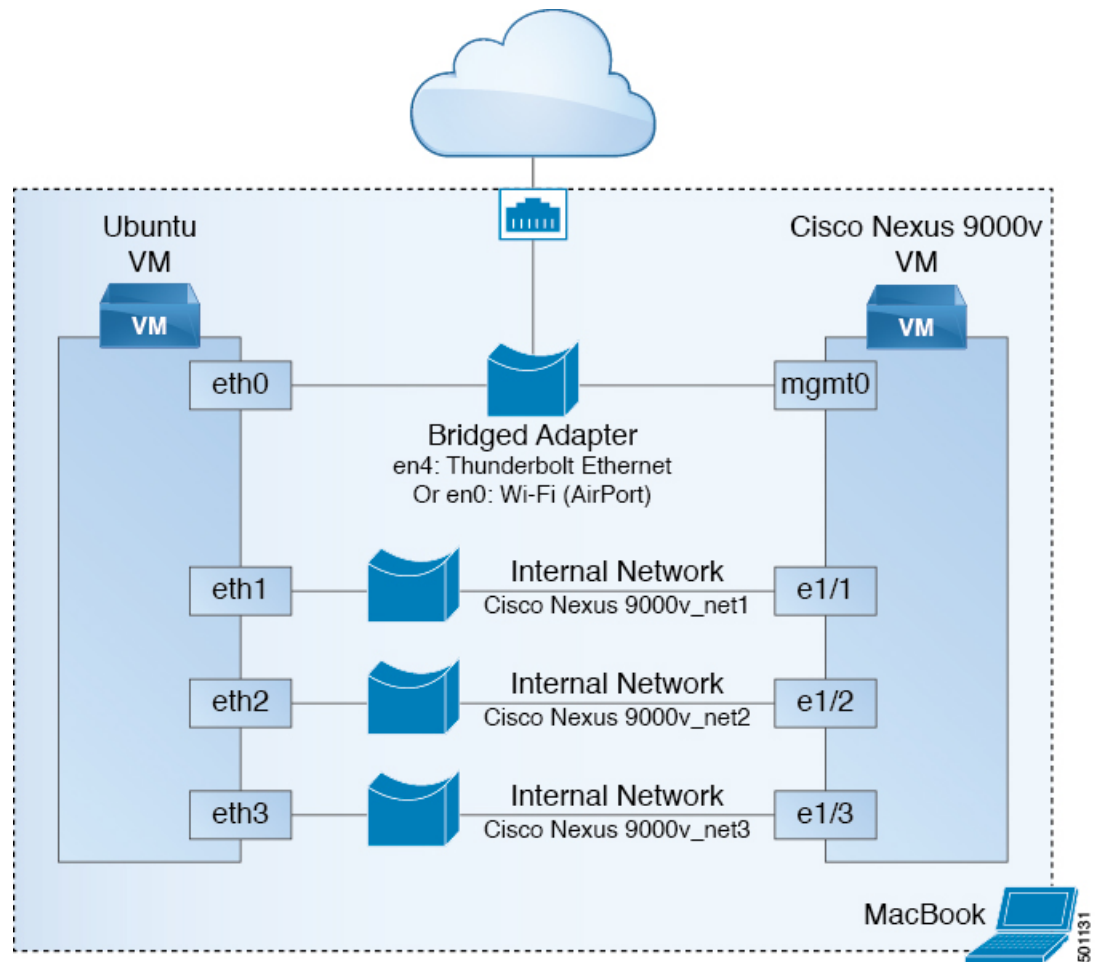
VM インスタンスを完全に削除する場合は、次を使用します。

```
$ vagrant box remove base
Removing box 'base' (v0) with provider 'virtualbox'...
$ vagrant destroy
default: Are you sure you want to destroy the 'default' VM? [y/N]
y
==> default: Destroying VM and associated drives..
```

## プラットフォームのインターコネクト

Nexus 9300v と他の仮想プラットフォーム間のネットワークは、VBox 内部ネットワークに基づいています。次の接続図を参照してください。

図 2: Vagrant VM を介した Cisco Nexus 9000v プラットフォームのインターコネクト



## イメージアップグレードのワークフロー

このセクションでは、Cisco Nexus 9000v プラットフォームの一般的なアップグレード手順について説明します。

### 新しいアーティファクトからの展開

環境に応じて、適切な仮想アーティファクトを使用し、次のセクションのいずれかを参照して VM を展開します。

- [KVM/QEMU に対する Nexus 9000v 展開ワークフロー \(25 ページ\)](#)
- [ESXi の Nexus 9000v 展開ワークフロー \(31 ページ\)](#)
- [Vagrant 用 Nexus 9000v 展開ワークフロー \(33 ページ\)](#)

## 新しい NX-OS イメージからのアップグレード

Nexus 9300v のアップグレードは、Cisco Nexus 9000v リリース 9.3(1) 以降の仮想アーティファクトで作成された VM からのみ許可されます。アップグレードする前に、ブートフラッシュに 400Mb 以上の新しい NX-OS バイナリ イメージがあることを確認してください。アップグレードするには、新しいバイナリをブートフラッシュにコピーしてから、標準の NX-OS ワークフローを使用してアップグレードします (例: `'install all nxos bootflash:///<nxos.bin>'`)。

これはプラットフォームの最初のリリースであるため、Nexus 9500v のアップグレードはサポートされていません。

Nexus 9300v および 9500v lite の場合、以前のバイナリ イメージから Lite バイナリ イメージへの ISSU はサポートされていません。コールドブートを使用してイメージを起動できる場合でも、最初に以前の構成を削除してから、lite バイナリをインストールします。





## 第 4 章

# Nexus 9300v および 9500v Lite NX-OS イメージ

- [Nexus 9300v および 9500v Lite NX-OS イメージについて \(41 ページ\)](#)
- [サポートされている機能 \(43 ページ\)](#)
- [Nexus 9300v Lite および 9500v Lite NX-OS イメージの展開 \(45 ページ\)](#)

## Nexus 9300v および 9500v Lite NX-OS イメージについて

NX-OS Lite (nxos64-cs-lite.10.5.3.F.bin) image のメモリ フットプリントが縮小され、Cisco Nexus 9300v および 9500v の以前のイメージよりもはるかに小さいイメージサイズになります。このイメージは、起動時にインストールでき、以前のイメージよりも速く起動します。ただし、一部の構成が有効でない可能性があるため、Lite イメージをロードする前に、write-erase-reload を実行します。



- (注) このイメージには、IPv4/IPv6 アンダーレイ (OSPF/IS-IS) を使用した、VXLAN EVPN、F+L なし、および BL 上の vPC、PC/vPC、LACP、および DME/YANG/テレメトリを含む M サイト機能などの展開シナリオをサポートするために必要な基本 RPM のみが含まれています。残りの RPM は、Lite NX-OS イメージから除外されます。

## リソース要件

次の表は、Nexus 9300v および 9500v Lite のリソース要件を示しています。

Resource	仕様
最小構成の RAM	4.5 G (基本ブートアップ)
推奨 RAM	8 G (機能の数による)

Resource	仕様
最小 vCPU	2
推奨される vCPUs	4
最小 vNIC	1
Maximum vNICs	65

## 仮想アーティファクト

次の表に、Nexus 9300v および 9500v Lite の仮想アーティファクトを示します：

NIC のタイプ	ハイパーバイザ	仮想アーティファクト	説明
N9300v Lite	ESXI 8.0	nexus9300v64-lite.10.5.1.Fova	仮想ディスク、マシン定義、およびNXOSイメージが含まれています。
N9300v Lite	KVM/QEMU 4.2.0	nexus9300v64-lite.10.5.1.Fqcow2	ブートフラッシュに仮想ディスクとNXOSイメージが含まれています。
N9300v Lite	Vagrant 2.3.7	nexus9300v64-lite.10.5.1.Fbox	マシン定義とともに仮想ディスクにプレインストールされたNXOSイメージが含まれています。
N9500v Lite	ESXI 8.0	nexus9500v64-lite.10.5.1.Fova	仮想ディスク、マシン定義、およびNXOSイメージが含まれています。
N9500v Lite	KVM/QEMU 4.2.0	nexus9500v64-lite.10.5.1.Fqcow2	ブートフラッシュに仮想ディスクとNXOSイメージが含まれています。

## サポートされている機能

次のセクションでは、Nexus 9300v Lite および 9500v Lite プラットフォームでサポートされる機能のさまざまなカテゴリについて説明します。

- [レイヤ 2 の機能](#)
- [レイヤ 3 の機能](#)
- [プログラマビリティ機能](#)

### レイヤ 2 の機能

次の表に、Nexus 9300v Lite および Nexus 9500v Lite プラットフォームのレイヤ 2 機能のサポートを示します。

機能	Nexus 9300v Lite のサポート	Nexus 9500v Lite のサポート
802.1AB LLDP	○	○
802.1Q VLAN/トランク	○	○
802.1s RST	○	○
802.3ad LACP	○	○
L2 マルチキャスト	はい（ブロードキャストとして）	はい（ブロードキャストとして）
MLAG	○	○
ポート チャンネル	○	○
VLAN	○	○
VXLAN EVPN	○	○
IPv4/IPv6 アンダーレイ（OSPF/IS-IS）で F+L なし	○	○
BGW 上の非 vPC を使用したマルチサイト機能	○	○
HSRP	○	○
NGOAM	○	○
TACACS+	○	○

### レイヤ 3 の機能

次の表に、Nexus 9300v Lite および Nexus 9500v Lite プラットフォームのレイヤー 3 機能のサポートを示します。

機能	Nexus 9300v Lite のサポート	Nexus 9500v Lite のサポート
OSPF	○	○
OSPFv3	○	○
BGP	○	○
MP-BGP	○	○
IS-IS	はい（ブロードキャストとして）	はい（ブロードキャストとして）
等価コスト マルチパス ルーティング（ECMP）	○	○
PIM-SM	○	○
MPLS	○	○
CDP	○	○
L3 SVI	○	○
サブインターフェイス	○	○
IPsec	非対応	非対応

### プログラマビリティ機能

次の表に、Nexus 9300v Lite および Nexus 9500v Lite プラットフォームのプログラマビリティ機能のサポートを示します。

機能	Nexus 9300v Lite のサポート	Nexus 9500v Lite のサポート
Bash シェル アクセスとスクリプティング	○	○
RPM サポート	○	○
システム状態へのプログラムによるアクセス (Python)	○	○
OS 内の Docker	○	○
NXAPI	○	○

機能	Nexus 9300v Lite のサポート	Nexus 9500v Lite のサポート
DME	○	○
RESTCONF	○	○
NETCONF	○	○
YANG モデル	○	○
テレメトリ	○	○
GNMI	○	○
NxSDK	○	はい

## Nexus 9300v Lite および 9500v Lite NX-OS イメージの展開

KVM/QEMU、ESXi、および Vagrant の展開手順の詳細については、次を参照してください。

- [ESXi の Nexus 9000v 展開ワークフロー \(31 ページ\)](#)
- [KVM/QEMU に対する Nexus 9000v 展開ワークフロー \(25 ページ\)](#)
- [Vagrant 用 Nexus 9000v 展開ワークフロー \(33 ページ\)](#)



(注) 9500v Lite は Vagrant ではサポートされていません。さらに、ISSU は 9300v Lite および 9500v Lite ではサポートされていません。詳細については、「[イメージアップグレードのワークフロー \(38 ページ\)](#)」を参照してください。





## 付 録 **A**

# Cisco Nexus 9000v のトラブルシューティング

---

この章は、次の項で構成されています。

- [Cisco Nexus 9000v プラットフォームのトラブルシューティング \(47 ページ\)](#)
- [Cisco Nexus 9000v データプレーンのトラブルシューティング \(53 ページ\)](#)

## Cisco Nexus 9000v プラットフォームのトラブルシューティング

### 一般的なトラブルシューティング/デバッグ

次の CLI コマンドは、Nexus 9300v と Nexus 9500v の両方のプラットフォームのトラブルシューティング ヘルプを提供します。

#### **show tech-support nexus9000v**

このコマンドの出力例を次に示します。

```
switch# show tech-support nexus9000v

----- Virtual Chassis Manager Debugs -----

#####
# /cmn/pss/virt_cmgr.log
#####
[19-12-10 20:42:34.160609]: virt_cmgr_startup_init called
[19-12-10 20:42:34.161351]: virt_cmgr_validate_file returned success
[19-12-10 20:42:34.161390]: Version 1, VNIC_scheme 2
[19-12-10 20:42:34.161404]: VM sup1: Module no 26, upg_version 1, type 1, card_i
ndex 0, image loc None
...
...
...
```

## すべてのハイパーバイザに共通の問題

### VM が「loader>」プロンプトに落ちたときに起動する

通常、初期ブートは成功します。ただし、VM のプロビジョニング方法によっては、システムブートが失敗し、VGA コンソールまたはシリアルコンソールに「loader>」プロンプトが表示される可能性があります。

例：

```
Loader Version 5.9

Loader > dir

bootflash::

    .rpmstore
    nxos.9.3.2.20.bin
    bootflash_sync_list
    .swtam
    eem_snapshots
    virtual-instance
    scripts
    platform-sdk.cmd

loader > boot nxos.9.3.2.20.bin
```

ブートを続行するには、「loader>」プロンプトで **boot nxos.9.3.2.20.bin** コマンドを入力します。

### VM が「loader>」プロンプトにドロップしないようにする

Cisco Nexus 9000v をセットアップした後 (および POAP インターフェイスのセットアップに続いて)、システムのブートイメージを設定して、リロード/シャットダウン後に「loader>」プロンプトにドロップしないようにします。

例：

```
nx-osv9000-2# config t
Enter configuration commands, one per line. End with CNTL/Z.
nx-osv9000-2(config)# boot nxos bootflash:nxos.9.3.2.20.bin
Performing image verification and compatibility check, please wait....
nx-osv9000-2(config)# copy running-config startup-config
```

### ブートアップ警告メッセージ

ブートアップ中に、次のような警告メッセージが表示される場合があります。

```
Checking all filesystems. **Warning** : Free memory available in bootflash is
553288 bytes
need at least 2 GB space for full image installation ,run df -h
```

このメッセージは通常、Nexus 9000v ブートフラッシュに別のイメージを保持するための十分なメモリ スペースがないことを示しています。この警告メッセージをなくすには、ブートフラッシュ スペースを解放して、別のバイナリ イメージをダウンロードできるようにします。

### Nexus 9000v Mac エンコード モード ネットワーク マッピング チェック

このチェックは、Nexus 9500v プラットフォームで **platform vnic scheme mac-encoded** コマンドを明示的に入力した場合にのみ関係します。このコマンドは、vNIC Mac エンコード スキームを有効にします。いずれかのデータトラフィックが通過する場合、または vNIC にマッピングされたインターフェイスに「リンクが接続されていません」状態が表示される場合は、Nexus 9000v 情報 show コマンドを参照して、正しい vNIC マッピングを確認します。

## ESXi ハイパーバイザの問題

### VM の電源を入れた後、Nexus 9000v のブートが表示されない

この問題の考えられる原因は、EFI ブートが VM 構成で設定されていないことです。この問題を解決するには、分散 OVA 仮想アーティファクトを使用したデプロイ後に、ESXi 展開ガイドを参照して、[仮想マシン設定の編集 (Edit virtual machine settings)] > [VM オプション (VM Options)] > [ブート オプション (Boot Options)] で「BIOS」を「EFI」に変更します。

### VGA 出力後に起動ログが表示されない

ESXi のブートアップ中によくある問題は、VGA コンソールに次のような出力が表示されることです。

```
Sysconf checksum failed. Using default values
console (dumb)

Booting nxos.9.3.2.6.bin...
Booting nxos.9.3.2.bin
Trying diskboot
  Filesystem type is ext2fs, partition type 0x83
Image valid

Image Signature verification for Nexus9000v is not performed.

Boot Time: 12/5/2019 10:38:41
```

問題は、VGA コンソールでは、起動プロセスに次のアクティビティがないことです。スイッチブートアッププロセスがハングすると誤解されることがよくあります。スイッチブートアップの出力を表示するには、ESXi ハイパーバイザ展開ガイドに記載されている手順に基づいて、プロビジョニングされたシリアル コンソールに接続します。

シリアル コンソールで何も起こらない場合、または「telnet: リモート ホストに接続できません: 接続が拒否されました」というエラー メッセージが表示される場合は、次の 1 つ以上の問題を示しています。

- VM 構成のシリアル コンソール プロビジョニングが正しくありません。ESXi 導入ガイドのシリアル コンソール接続の手順を読んで、それに従ってください。
- サポートされているバージョンは ESXi 8.0 展開のみです。ESXi vCenter の有効なライセンスと有効な UCS サーバー ライセンスがあることを確認してください。
- サーバーの「セキュリティプロファイル」に、着信接続と発信接続の両方で「ネットワーク経由で接続された VM シリアル ポート」があることを確認します。

### VM の電源を切った後、「loader>」プロンプトにアクセスできない

この問題は、VM をパワーオンして期待どおりに起動したが、シリアルコンソールが正しくプロビジョニングされなかった場合に発生します。次に、「`config t; boot nxos bootflash:nxos.9.3.2.20.bin`」`configure` が実行され、保存されます。VM の電源を再度入れると、VGA コンソールにドロップします。

次の推奨事項は、ESXi ハイパーバイザーでこの問題を回避するのに役立ちます。

EFI BIOS は、VM コンソールへのすべての入出力をデフォルトで設定します。VM が「loader>」プロンプトにドロップしたら、vSphere クライアントまたは VGA コンソールに移動して、「loader>」プロンプトにアクセスして、ハードディスク内のイメージを起動します。この動作を変更するには、ESXi VM 編集モードで追加の構成を追加します。次のいずれかの方法を使用します。

1. vSphere クライアントの [構成パラメータ] ウィンドウで、構成に 1 行追加します ([設定の編集]>[VM オプション]>[詳細]>[構成の編集])。
2. VM が作成されたら、`efi.serialconsole.enabled = "TRUE"` を .vmx ファイルに追加します。

### Cisco Nexus 9000v が起動するとすぐに vCenter または UCS サーバーの接続が失われる



**注意** vNIC を vSwitch またはブリッジに接続する場合、ネットワーク接続が正しくないと、ハイパーバイザー サーバーまたは ESXi 上の vCenter への接続が失われることがあります。

Cisco Nexus 9000v は、ESXi のグラフィック表現から入力された vNIC を、ハイパーバイザーサーバー内の外部または内部のネットワークに使用します。最初の NIC は、常に Cisco Nexus 9000v 管理インターフェイスとして使用されます。

Cisco Nexus 9000v VM の最初の NIC は管理インターフェイスです。ラボの LAN 物理スイッチまたは vSwitch (VM ネットワーク) に直接接続します。サーバー管理接続と競合する物理スイッチにデータ ポート vNIC を接続しないでください。

### Cisco Nexus 9000v データ ポートが ESXi サーバーでトラフィックを渡さない

スムーズな操作を確保するには、vSwitch の特定の構成設定を有効にする必要があります。

- Cisco Nexus 9000v に接続している vSwitch のすべてのインスタンスが「無差別モード」= 「受け入れ」になっていて、UCS サーバを指していることを確認します。このオプションには、vSphere Client から [構成]>[プロパティ]>[編集] からアクセスできます。
- vSwitch のすべてのインスタンスがすべての VLAN を通過するようにします。このオプションには、vSphere Client から [構成 (Configuration)]>[プロパティ (Properties)]>[編集 (Edit)] でアクセスできます。

ESXi 8.0 ハイパーバイザーは、多くの場合、ネットワーク インターフェイス アダプターを、Nexus 9000v プラットフォームでサポートされていない「E1000E」タイプにデフォルト設定し

ます。展開後、すべてのネットワーク アダプターの種類が「E1000」であることを確認します。

## KVM/QEMU ハイパーバイザの問題

KVM/QEMU コマンドラインオプションを理解するには、基本的な Linux の知識が必要です。このハイパーバイザーに Nexus 9000v を展開するには、展開手順に従い、次の領域に注意してください。

- ユーザー ガイドで bios.bin が推奨されていることを確認してください。
- コマンドラインで複数のディスク入力サポートされている場合は、VMが他のデバイスから起動しないように、起動可能なディスクが `bootindex=1` に設定されていることを確認します。
- 複雑なコマンドラインを実装しようとしている場合は、基本的な KVM/QEMU 展開手順に従って、最初に単純なスイッチ インスタンスを起動してユーザー環境を確認します。

### KVM または QEMU ハイパーバイザでのマルチキャスト

Cisco Nexus 9000v のマルチキャスト機能はブロードキャストとしてサポートされています。この機能を正しく動作させるには、この環境ですべてのブリッジインターフェイスで IGMP マルチキャスト スヌーピングを無効にします。

次の例は、Linux プロンプトから `vxlan_br1`、`vxlan_br2`、`vxlan_br3`、および `vxlan_br4` を無効にする方法を示しています。

```
echo 0 > /sys/devices/virtual/net/vxlan_br1/bridge/multicast_snooping
echo 0 > /sys/devices/virtual/net/vxlan_br2/bridge/multicast_snooping
echo 0 > /sys/devices/virtual/net/vxlan_br3/bridge/multicast_snooping
echo 0 > /sys/devices/virtual/net/vxlan_br4/bridge/multicast_snooping
```

LLDP、LACP などの L2 パケットを渡すには、KVM/QEMU 導入ガイドの Linux ブリッジマスクの設定に従ってください。

## Vagrant/VirtualBox の問題

### VirtualBox/Vagrant でのネットワーキング

VirtualBox/Vagrant で dataplane インターフェイスを使用するには、次のことを確認してください。

- インターフェイスが「無差別（promiscuous）」モードである必要があります。
- VirtualBox のネットワーク設定で、無差別モードに対し「すべて許可（Allow All）」を選択します。
- `show interface mac` コマンドを使用して、トポロジ内の Cisco Nexus 9000v のすべてのインスタンスに一意の MAC アドレスがあることを確認します。

VirtualBox/Vagrant での VM の通常の起動:

```
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
The configured shell (config.ssh.shell) is invalid and unable
to properly execute commands. The most common cause for this is
using a shell that is unavailable on the system. Please verify
you're using the full path to the shell and that the shell is
executable by the SSH user.
```

通常のブートアップが成功した後、**vagrant ssh** コマンドは Nexus 9000v スイッチプロンプトにアクセスします。

以下は、VM のブートアップに失敗する可能性のある 1 つの例です。

```
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'base'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: n9kv31_default_1575576865720_14975
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
Timed out while waiting for the machine to boot. This means that
Vagrant was unable to communicate with the guest machine within
the configured ("config.vm.boot_timeout" value) time period.
```

If you look above, you should be able to see the error(s) that Vagrant had when attempting to connect to the machine. These errors are usually good hints as to what may be wrong.

If you're using a custom box, make sure that networking is properly working and you're able to connect to the machine. It is a common problem that networking isn't setup properly in these boxes. Verify that authentication configurations are also setup properly, as well.

If the box appears to be booting properly, you may want to increase the timeout ("config.vm.boot\_timeout") value.

障害をトラブルシューティングするには、次の手順を実行します。

- メモリやvCPUなどの十分なリソースが使用可能であることを確認します。PCまたはサーバーで大量のメモリを消費するすべてのアプリケーションを閉じます。使用可能な空きメモリを確認してください。
- **vagrant halt -f** を入力して VM の電源を切ります。

- VM の電源を切った後、VirtualBox GUI に移動します。VM シリアル コンソールを有効にして、起動プロセスを監視し、[ポート]->[シリアルポートを有効にする]で考えられる問題を表示します。

または、次の VBox コマンドを使用して、このゲスト シリアル コンソールを有効にします。VM 名を見つけます。

```
VBoxManage list vms
    "n9kv_default_1575906706055_2646" {0b3480af-b9ac-47a4-9989-2f5e3bdf263f}
```

次に、シリアル コンソールを有効にします。

```
VBoxManage modifyvm n9kv_default_1575906706055_2646 --uart1 0x3F8 4
```

- 元の「vagrant up」を入力した同じ端末から「vagrant up」と入力して、VM の電源を再度オンにします。
- シリアルコンソールにアクセスするには、コンピューターの別の端末から「telnet localhost 2023」と入力します。
- シリアル コンソールからの出力を観察して、ブートアップの問題を確認します。
- ゲストシリアルコンソールが不要になった場合は、シリアルコンソールをオフにします。次の VBox コマンドを使用するか、VirtualBox GUI 設定に移動して [シリアル ポートを有効にする] の選択を解除します。

```
VBoxManage modifyvm n9kv_default_1575906706055_2646 --uart1 off
```

## Cisco Nexus 9000v データプレーンのトラブルシューティング

このセクションの debug コマンドと show コマンドは、Cisco Nexus 9300v と Cisco Nexus 9500v の両方のプラットフォームのトラブルシューティングに使用できます。これらのコマンドは、ラインカード/モジュールで実行する必要があります。

### debug コマンド

- debug l2fwder event
- debug l2fwder error
- debug l2fwder fdb
- debug l2fwder pkttrace

これらのコマンドのいずれかを実行するには、次の例に従ってラインカードに接続します。

```
switch# sh mod | inc Mod
Mod Ports      Module-Type      Model      Status
1    64    Nexus 9000v 64 port Ethernet Module  N9K-X9364v  ok
27   0    Virtual Supervisor Module  N9K-vSUP    active *
```

```
Mod Sw          Hw      Slot
```

```

Mod  MAC-Address(es)                               Serial-Num
Mod  Online Diag Status
switch# attach mod 1
Attaching to module 1 ...
To exit type 'exit', to abort type '$.'
module-1# debug l2fwder ?
  error      Configure debugging of l2fwder control and data path errors
  event      Configure debugging of l2fwder events over ipc
  fdb        Configure debugging of l2fwder events over fdb
  ha         Configure debugging of l2fwder events from sysmgr
  logfile    Enable file logging to /logflash/l2fwder.debug
  packet     Configure debugging of l2fwder packet forwarding information
  pkttrace   Configure debugging of l2fwder packet trace

module-1# debug l2fwder

```

## Event History コマンド

- show system internal l2fwder event-history events
- show system internal l2fwder event-history errors
- show system internal l2fwder event-history fdb

## コマンドの表示

### show system internal l2fwder table bd

```

v-switch# show system internal l2fwder table bd

vlan   1  member 3, 4, 5,  untagged 3, 4, 5,  STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan no
vlan  80  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan  90  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 110  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 210  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 310  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 410  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 510  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 550  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan no
vlan 560  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan no
vlan 610  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes
vlan 650  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan no
vlan 660  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan no
vlan 710  member 3, 4, 5,  untagged none      STP ports 3, 4, 5,  dis none  blk_lis none
         lrn none  fwd 3, 4, 5,  tid 1, 2,  vxlan yes

```

```
vlan 810 member 3, 4, 5, untagged none STP ports 3, 4, 5, dis none blk_lis none
lrn none fwd 3, 4, 5, tid 1, 2, vxlan yes
```

### show system internal l2fwder table if

```
v-switch# show system internal l2fwder table if
```

If_name	If_index	gport	fd	untagged	vlanid	Trunk	SVP Info	Native
Ethernet1/1	0x1a000000	0x8000801	14	1	4095	0x0	none	4095
Ethernet1/2	0x1a000200	0x8000802	15	1	4095	0x0	none	4095
Ethernet1/3	0x1a000400	0x8000803	16	0	4045	0x1	none	40451
Ethernet1/4	0x1a000600	0x8000804	17	0	810	0x2	none	810
Ethernet1/5	0x1a000800	0x8000805	18	0	810	0x0	none	810
Ethernet1/6	0x1a000a00	0x8000806	0	1	4095	0x0	none	4095
Ethernet1/7	0x1a000c00	0x8000807	0	1	4095	0x0	none	4095
Ethernet1/8	0x1a000e00	0x8000808	0	1	4095	0x0	none	4095
Ethernet1/9	0x1a001000	0x8000809	0	1	4095	0x0	none	4095
Ethernet1/10	0x1a001200	0x800080a	0	1	4095	0x0	none	4095
Ethernet1/11	0x1a001400	0x800080b	0	1	4095	0x0	none	4095

### show system internal l2fwder table port-channel

```
v-switch# show system internal l2fwder table port-channel
```

Port-channel	Count	Member-list
0x1	1	0x8002004
0x4	2	0x8005001 0x8000805
0x5	2	0x8002001 0x8000801

Port-channel	Count	Local member-list6
0x1	0	
0x4	1	0x8000805
0x5	1	0x8000801

### show system internal l2fwder table vxlan peer

```
v-switch# show system internal l2fwder table vxlan peer
```

```
VXLAN Tunnel:
  src_ip: 6.6.6.6, Is VxLAN enabled = TRUE
  multisite: no, nve_tun_dci_sip: 0.0.0.0
VXLAN PEER: No of tunnels = 7
  peer_ip: 224.1.1.2, vxlan_port_id: 0x0,
    tunnel_id: 0x4c000000, is_dp: 0 is_dci: 0
  peer_ip: 224.1.1.4, vxlan_port_id: 0x0,
    tunnel_id: 0x4c000002, is_dp: 0 is_dci: 0
  peer_ip: 224.1.1.6, vxlan_port_id: 0x0,
    tunnel_id: 0x4c000004, is_dp: 0 is_dci: 0
  peer_ip: 224.1.1.8, vxlan_port_id: 0x0,
    tunnel_id: 0x4c000006, is_dp: 0 is_dci: 0
  peer_ip: 224.1.1.9, vxlan_port_id: 0x0,
    tunnel_id: 0x4c000008, is_dp: 0 is_dci: 0
  peer_ip: 224.1.1.10, vxlan_port_id: 0x0,
    tunnel_id: 0x4c00000a, is_dp: 0 is_dci: 0
  peer_ip: 6.5.5.5, vxlan_port_id: 0x80002db8,
    tunnel_id: 0x4c00050a, is_dp: 0 is_dci: 0
Tunnel_id entry:
  peer_ip: 224.1.1.2, tunnel_id: 0x4c000000
  peer_ip: 224.1.1.4, tunnel_id: 0x4c000002
```

```

peer_ip: 224.1.1.6, tunnel_id: 0x4c000004
peer_ip: 224.1.1.8, tunnel_id: 0x4c000006
peer_ip: 224.1.1.9, tunnel_id: 0x4c000008
peer_ip: 224.1.1.10, tunnel_id: 0x4c00000a
peer_ip: 6.5.5.5, tunnel_id: 0x4c00050a
Vxlan_gport ucast-entry:
peer_ip: 6.5.5.5, vxlan_port_id: 0x80002db8

```

### show system internal l2fwder table vxlan vni

```
v-switch# show system internal l2fwder table vxlan vni
```

VNI	VLAN	DF
----	----	----
81000	810	no
51000	510	no
5001	1001	no
5002	1002	no
5003	1003	no
5004	1004	no
21000	210	no
71000	710	no
9000	90	no
41000	410	no
11000	110	no
61000	610	no
31000	310	no

### show system internal l2fwder table if

```
v-switch# show system internal l2fwder acl info
```

Inactive List:

Entry ID: 14596 Qualify: DstTrunk 4, Action: RedirectTrunk 5 Prio: 4

Active List:

Inactive List:

Active List:

Entry ID: 15873 Qualify: EtherType ARP ForwardingVlanId 110, 610, 710, 1001, 1003,  
Action: CopyToCpu SET Drop SET Prio: 1

### show system internal l2fwder mac

```
v-switch# show system internal l2fwder mac
```

Legend:

\* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC  
+ - primary entry using vPC Peer-Link,  
(T) - True, (F) - False, C - ControlPlane MAC

	VLAN	MAC Address	Type	Secu	NTF	Del	Ports	Station_id
	-----	-----	-----	-----	-----	-----	-----	-----
*	1	008b.860d.1b08	static	F	F	0	0xc000005	0
G	-	008b.860d.1b08	static	F	F	0	sup-eth1(R)	508,
*	210	0000.4545.6767	dynamic	F	F	0	0xc000004	0
G	710	008b.bc90.1b08	static	F	F	0	sup-eth1(R)	0
G	310	008b.bc90.1b08	static	F	F	0	sup-eth1(R)	0
G	-	0002:0002:0002	static	F	F	0	sup-eth1(R)	1,

```

* 210 008b.860d.1b08 static F F 0 0xc000005 0
G 410 008b.bc90.1b08 static F F 0 sup-eth1 (R) 0
* 1003 008b.2b34.1b08 dynamic F F 1 nve (0x80002db9) 0
* 1002 008b.2b34.1b08 dynamic F F 1 nve (0x80002db9) 0
* 1001 008b.2b34.1b08 dynamic F F 1 nve (0x80002db9) 0
* 1004 008b.2b34.1b08 dynamic F F 1 nve (0x80002db9) 0
* 810 008b.860d.1b08 static F F 0 0xc000005 0
G 510 008b.bc90.1b08 static F F 0 sup-eth1 (R) 0
* 610 008b.2b34.1b08 dynamic F F 1 nve (0x80002db9) 0
G 1 008b.bc90.1b08 static F F 0 sup-eth1 (R) 0
G - 008b.bc90:1b08 static F F 0 sup-eth1 (R) 511,

```

### show system internal l2fwder table if

```
v-switch# show system internal l2fwder port egress info
```

```

Ingress port :          Blocked egress ports
+-----+-----+
0x8002001      1          5
0x8000801      1          5
0x8020821      1          5

```

### show system internal l2fwder vpc info

```
v-switch# show system internal l2fwder vpc info
```

```
VPC role : Primary
```

## パケット キャプチャ コマンド

Cisco Nexus N9000v は、スタンドアロンの Nexus 9000 ハードウェア スイッチと同様に Ethalyzer をサポートします。



## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。