

NSX-T インストール ガイド

変更日：2018 年 6 月 05 日

VMware NSX-T Data Center 2.2



vmware®

VMware Web サイトで最新の技術ドキュメントをご確認いただけます。

<https://docs.vmware.com/jp/>

VMware の Web サイトでは、最新の製品アップデートを提供しています。

本書に関するご意見、ご要望をお寄せください。フィードバック送信先：

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

ヴィエムウェア株式会社
105-0013 東京都港区浜松町 1-30-5
浜松町スクエア 13F
www.vmware.com/jp

Copyright © 2018 VMware, Inc. All rights reserved. 著作権および商標.

内容

NSX-T インストール ガイド 5

1 NSX-T の概要 6

- 管理プレーン 7
- 制御プレーン 9
- データ プレーン 10
- 論理スイッチ 11
- 分散論理ルーター 11
- 用語の説明 12

2 インストールの準備 16

- システム要件 16
- ポートとプロトコル 20
- NSX-T インストールの高水準のタスク 26

3 KVM の使用 28

- KVM のセットアップ 28
- KVM CLI を使用したゲスト仮想マシンの管理 33

4 NSX Manager のインストール 35

- NSX Manager および利用可能なアプライアンスのインストール 37
- コマンドライン OVF ツールを使用した ESXi への NSX Manager のインストール 39
- KVM への NSX Manager のインストール 42
- 新しく作成された NSX Manager にログインします。 44

5 NSX Controller のインストールとクラスタリング 46

- NSX Manager からのコントローラとクラスタの自動インストール 48
- グラフィカル ユーザー インターフェイス (GUI) を使用した ESXi への NSX Controller のインストール 55
- コマンドライン OVF ツールを使用した ESXi への NSX Controller のインストール 57
- KVM への NSX Controller のインストール 59
- NSX Manager への NSX Controller の追加 62
- コントロール クラスタの初期化によるコントロール クラスタ マスターの作成 63
- クラスタ マスターを使用した NSX Controller の追加 65

6 NSX Edge のインストール 69

- NSX Edge のネットワーク設定 71
- NSX Manager からの NSX Edge 仮想マシンの自動展開 77
- vSphere のグラフィカル ユーザー インターフェイス (GUI) を使用した ESXi への NSX Edge のインストール 79

- コマンドライン OVF ツールを使用した ESXi への NSX Edge のインストール 81
- PXE サーバで ISO ファイルを使用した NSX Edge のインストール 84
- ベア メタルへの NSX Edge のインストール 91
- ISO ファイルを使用した仮想アプライアンスとしての NSX Edge のインストール 92
- NSX Edge の管理プレーンへの追加 95

7 ホストの準備 97

- KVM ホストへのサードパーティ製パッケージのインストール 97
- RHEL KVM ホストの Open vSwitch のバージョンを確認する 99
- NSX-T ファブリックへのハイパーバイザー ホストの追加 100
- NSX-T カーネル モジュールの手動インストール 105
- ハイパーバイザー ホストの管理プレーンへの追加 110

8 トランスポート ゾーンとトランスポート ノード 113

- トランスポート ゾーンについて 113
- 拡張データ パス 115
- トンネル エンドポイントの IP アドレス用 IP アドレス プールの作成 116
- アップリンク プロファイルの作成 119
- トランスポート ゾーンの作成 123
- ホスト トランスポート ノードの作成 125
- Network I/O Control の設定 142
- NSX Edge トランスポート ノードの作成 151
- NSX Edge クラスタの作成 155

9 NSX Cloud コンポーネントのインストール 156

- NSX Cloud のアーキテクチャとコンポーネント 156
- NSX Cloud コンポーネントのインストール 158

10 NSX-T のアンインストール 169

- NSX-T オーバーレイの設定解除 169
- NSX-T からのホストの削除、または NSX-T の完全なアンインストール 169

NSX-T インストール ガイド

『NSX-T インストール ガイド』では、VMware NSX-T[®] 製品をインストールする方法について説明します。詳細な設定手順や推奨されるベスト プラクティスについても記載しています。

対象読者

この情報は、NSX-T をインストールまたは使用するユーザーを対象としています。システム管理者としての経験があり、仮想マシン テクノロジーとネットワーク仮想化の概念に詳しい方を対象にしています。

VMware の技術ドキュメントの用語集

VMware は、新しい用語を集めた用語集を提供しています。VMware の技術ドキュメントで使用されている用語の定義については、<http://www.vmware.com/support/pubs> をご覧ください。

NSX-T の概要

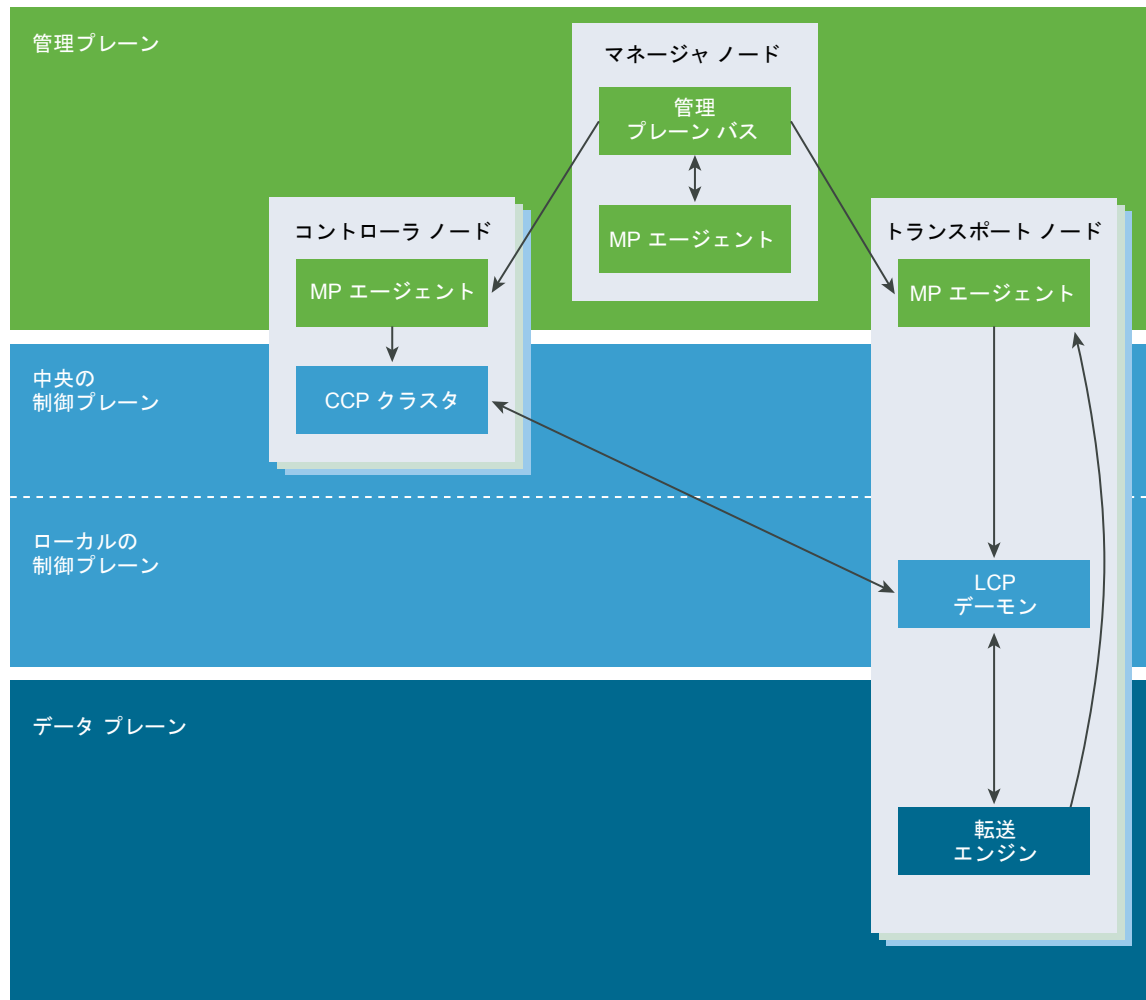
サーバ仮想化ではプログラムによって、ソフトウェア ベースの仮想マシンの作成、削除、リストア、およびスナップショットの作成を行います。NSX-T のネットワーク仮想化は、同じような方法で、ソフトウェア ベースの仮想ネットワークを作成、削除、リストアを行います。

ネットワーク仮想化では、ネットワーク ハイパーバイザーと同等の機能によって、レイヤー 2 からレイヤー 7 までのネットワーク サービス（スイッチング、ルーティング、アクセス コントロール、ファイアウォール、QoS など）をソフトウェアで完全に再現します。プログラムでこれらのサービスを任意に組み合わせ、独自の隔離された仮想ネットワークをわずか数秒で構築できます。

NSX-T は、管理プレーン、制御プレーン、およびデータ プレーンの 3 つのプレーンを実装することで機能します。それぞれ独立し、相互に連携する 3 つのプレーンは、管理ノード、制御ノード、およびトランスポート ノードの 3 種類のノードに、プロセス、モジュール、およびエージェントのセットとして実装されます。

- すべてのノードで管理プレーン エージェントをホストします。
- NSX Manager ノードは API サービスをホストします。NSX-T の各インストールでは、単一の NSX Manager ノードをサポートします。
- NSX Controller ノードは、統合制御プレーンのクラスタ デモンをホストします。
- NSX Manager および NSX Controller ノードは、同一の物理サーバ上でホストできます。

- トランスポート ノードは、ローカル制御プレーンのデーモンと転送エンジンをホストします。



この章には、次のトピックが含まれています。

- [管理プレーン](#)
- [制御プレーン](#)
- [データプレーン](#)
- [論理スイッチ](#)
- [分散論理ルーター](#)
- [用語の説明](#)

管理プレーン

管理プレーンはシステムへの単一の API エントリ ポイントで、ユーザー設定の維持とユーザー クエリの処理のほか、システム内の管理プレーン、制御プレーン、データプレーンのすべてのノードの操作を行います。

NSX-T では、ユーザー設定のクエリ、変更、維持に関するものはすべて管理プレーンの担当となり、その設定を適切なデータ プレーン要素に広めるのは制御プレーンの担当となります。これは、一部のデータが、その存在の段階に応じて、複数のプレーンに属することを意味します。管理プレーンは、制御プレーン、また場合によってはデータ プレーンへの最近のステータスや統計情報のクエリも処理します。

管理プレーンは、設定された（論理）システムの唯一の情報源であり、ユーザーが設定を通じて管理します。変更は、RESTful API または NSX-T のユーザー インターフェイスを使用して行います。

NSX には、すべてのコントローラ クラスタとトランスポート ノードで実行される管理プレーン エージェント (MPA) もあります。MPA はローカル アクセスとリモート アクセスが可能です。トランスポート ノードでは、データ プレーンに関連するタスクが実行される場合もあります。

管理プレーンでは次のタスクが実行されます。

- 設定のパーシステンス（適切な論理状態）
- 入力の検証
- ユーザー管理：ロールの割り当て
- ポリシー管理
- バックグラウンド タスクの追跡

NSX Manager

NSX Manager は、論理スイッチや NSX Edge Services Gateway などの NSX-T コンポーネントの作成、設定、監視を行うためのグラフィカル ユーザー インターフェイス (GUI) と REST API を提供する仮想アプライアンスです。

NSX Manager は、NSX-T エコシステムの管理プレーンです。NSX Manager は、NSX-T のネットワーク集中管理コンポーネントで、集約されたシステム ビューを提供します。後述の設定と連携が可能です。

- 論理ネットワーク コンポーネント：論理的なスイッチングとルーティング
- ネットワークと Edge サービス
- セキュリティ サービスと分散ファイアウォール

NSX Manager では、NSX-T で作成された仮想ネットワークに関連するワークロードの監視とトラブルシューティングの方法が利用できます。また、組み込みサービスと外部サービスとのシームレスなオーケストレーションが可能です。組み込みまたはサードパーティに関係なく、すべてのセキュリティ サービスが NSX-T の管理プレーンで展開、設定されます。管理プレーンでは、1 つの画面で複数のサービスの可用性を確認できます。また、ポリシー ベースのサービス チェーン、コンテキスト共有、サービス間イベントを容易に操作できます。これにより、セキュリティ状態の監査を簡素化し、ID ベースの制御（Active Directory やモビリティ プロファイルなど）を効率的に適用できるようになります。

NSX Manager は、コンポーネントの使用を自動化するための REST API エントリ ポイントにもなります。この柔軟なアーキテクチャにより、任意のクラウド管理プラットフォーム、セキュリティ ベンダー プラットフォーム、または自動化フレームワークを通じて、設定と監視に関するあらゆる要素を自動化できます。

NSX-T の管理プレーン エージェント (MPA) は、すべてのノード (ハイパーバイザー) に常駐する NSX Manager のコンポーネントです。MPA は、システムの適切な状態を維持し、また設定、統計、ステータス、リアルタイム データなどのフロー制御以外 (NFC) のメッセージをトランスポート ノードと管理プレーンの間でやりとりする役割を担います。

NSX Policy Manager

NSX Policy Manager は、インテント ベースのシステムを提供して NSX-T サービスの利用を簡素化する仮想アプライアンスです。

NSX Policy Manager には、ネットワーク、セキュリティ、可用性に関するインテントを指定するためのグラフィカル ユーザー インターフェイス (GUI) と REST API があります。

NSX Policy Manager は、ツリー ベースのデータ モデルの形式でユーザーからインテントを受け取り、そのインテントを実現するように NSX Manager を設定します。NSX Policy Manager は、NSX Manager で分散ファイアウォールを設定する通信インテント仕様をサポートします。

Cloud Service Manager

Cloud Service Manager (CSM) は、すべてのパブリック クラウド構築に、1 つの画面で管理できる管理エンドポイントを提供します。

CSM は、パブリック クラウド インベントリのオンボーディング、設定、および監視用のグラフィカル ユーザー インターフェイス (GUI) と REST API を提供する仮想アプライアンスです。

制御プレーン

管理プレーンからの構成に基づいてすべての短期的なランタイム状態を算出し、データ プレーン要素からレポートされたトポロジ情報を伝達し、ステートレス構成を転送エンジンにプッシュします。

制御プレーンは、NSX-T で 2 つの部分に分けられます。中央制御プレーン (CCP) は NSX Controller クラスタ ノードで実行され、ローカル制御プレーン (LCP) は制御対象のデータ プレーンに隣接するトランスポート ノードで実行されます。中央制御プレーンは、管理プレーンからの構成に基づいていくつかの短期的なランタイム状態を算出し、データ プレーン要素からレポートされた情報を、ローカル制御プレーンを介して伝達します。ローカル制御プレーンは、ローカル リンクの状態を監視し、データ プレーンおよび CCP から得た最新情報に基づいて最も短期的なランタイム情報を算出し、ステートレス構成を転送エンジンにプッシュします。LCP は、それをホストするデータ プレーン要素に依存します。

NSX Controller

中央制御プレーン (CCP) と呼ばれる NSX Controller は、仮想ネットワークとオーバーレイ トランスポート トンネルを制御する高度な分散状態管理システムです。

NSX Controller は、可用性に優れた仮想アプライアンスのクラスタとして展開され、NSX-T アーキテクチャ全体における仮想ネットワークをプログラムで展開する役割を担います。NSX-T の CCP はすべてのデータ プレーン トラフィックから論理的に分離されます。このため、制御プレーンで障害が発生しても、既存のデータ プレーンの処理に影響はありません。トラフィックはコントローラを経由しません。コントローラは、論理スイッチ、分散論理ルーター、Edge 設定など、他の NSX Controller コンポーネントに設定を提供する役割を担います。ネットワークでは、データ転送の安定性と信頼性が、重要な懸念事項です。高可用性と拡張性をさらに向上するために、NSX Controller は 3 インスタンスのクラスタで展開されます。

データ プレーン

制御プレーンによって入力されたテーブルに基づいて、パケットのステートレス転送/変換を行い、トポロジ情報を制御プレーンに報告して、パケット レベルの統計情報を保持します。

データ プレーンは、物理トポロジと状態、たとえば VIF の場所、トンネルの状態などの情報源です。パケットを 1 つの場所から別の場所に移動する処理を行っているのがデータ プレーンです。また、データ プレーンは、複数のリンク/トンネルの状態を管理し、フェイルオーバーを処理します。遅延やジッターの要件が非常に厳しい場合、パケット単位のパフォーマンスが重要です。データ プレーンはカーネル、ドライバ、ユーザースペース、または特定のユーザースペース プロセスに完全に含まれているとは限りません。データ プレーンは、制御プレーンによって入力されるテーブル/ルールに基づいて、完全にステートレスな転送に制約されます。

データ プレーンには、TCP ターミネーションなどの機能の状態を、ある程度まで保持するコンポーネントが存在する場合もあります。これは、MAC:IP アドレス トンネル マッピングなど、制御プレーンで管理される状態とは異なります。制御プレーンで管理される状態はパケットの転送方法に関するものであるのに対して、データ プレーンで管理される状態はペイロードの操作方法に限られます。

NSX Edge

NSX Edge は、ルーティング サービスと NSX-T 環境の外部のネットワークへの接続を提供します。

NSX Edge は、ベアメタル ノードまたは仮想マシンとして展開できます。

NSX Edge は、NSX-T ドメインから、Tier-0 ルーターを経由して、BGP またはスタティック ルーティングで外部接続を確立するために必要です。また、Tier-0 または Tier-1 のいずれかの分散論理ルーターでネットワーク アドレス変換 (NAT) サービスが必要な場合は、NSX Edge を展開する必要があります。

NSX Edge ゲートウェイは NAT、ダイナミック ルーティングなどの一般的なゲートウェイ サービスを提供して、分離されたスタブネットワークを共有（アップリンク）ネットワークへ接続します。NSX Edge は一般的に DMZ やマルチテナントのクラウド環境などに展開されますが、NSX Edge は各テナント用の仮想境界を構築します。

トランスポート ゾーン

トランスポート ゾーンは、ホスト論理スイッチが接続できるホストを制御する論理的な構造です。トランスポート ゾーンは 1 つ以上のホスト クラスタにまたがって設定できます。トランスポート ゾーンでは、特定のネットワークを使用できるホストと仮想マシンを指定します。

トランスポート ゾーンは、物理ネットワーク インフラストラクチャを介して相互に通信できるホストの集合を定義します。この通信は、仮想トンネル エンドポイント (VTEP) として定義されている、1 つ以上のインターフェイスを介して行われます。

トランスポート ノードは、ローカル制御プレーン デーモンを実行するホストです。また、NSX-T データ プレーンを実装する転送エンジンです。トランスポート ノードは、使用可能なネットワーク サービスの設定に応じてパケットを交換する NSX-T Distributed Switch (N-VDS) で構成されます。

2 台のトランスポート ノードが同じトランスポート ゾーンにある場合、両方のトランスポート ノードでホストされる仮想マシンは、同じトランスポート ゾーン内の NSX-T 論理スイッチを認識して、接続できます。これにより、仮想マシンがレイヤー 2/レイヤー 3 に到達できる場合は、仮想マシン同士が互いに通信できるようになります。各仮想マシンが、それぞれ別のトランスポート ゾーン内のスイッチに接続されている場合、それらの仮想マシンは互いに通信できません。トランスポート ゾーンは、レイヤー 2/レイヤー 3 接続性要件に変わるものではありませんが、接続性に制約を加えます。つまり、相互に接続するには、前提条件として同じトランスポート ゾーンに属する必要があります。この前提条件が満たされれば、相互接続は可能になりますが、自動的に通信が可能となるわけではありません。実際に接続を可能にするには、レイヤー 2 および別のサブネットの場合のレイヤー 3 ネットワークの設定と条件が正しく動作している必要があります。

ホストに 1 台以上の NSX 管理対象分散仮想スイッチ（以前はホスト スイッチと呼ばれていた N-VDS）が含まれている場合、ホストはトランスポート ノードとして機能します。ホスト トランスポート ノードを作成し、トランスポート ゾーンに追加すると、NSX-T によってホストに N-VDS がインストールされます。ホストが属する各トランスポート ゾーンごとに、個別に N-VDS がインストールされます。N-VDS は、仮想マシンを NSX-T 論理スイッチに接続するとき、および NSX-T 論理ルーターのアップリンクとダウンリンクを作成するときに使用されます。

論理スイッチ

NSX-T プラットフォームの論理スイッチング機能によって、仮想マシンと同じ柔軟性と俊敏性で、独立型の論理 L2 ネットワークを追加できます。

論理スイッチは、レイヤー 3 の IP アドレス アクセスが可能な多数のホストにわたるレイヤー 2 スイッチ接続を表します。論理ネットワークを一部のホストに制限するか、接続についてカスタムの要件がある場合は、追加の論理スイッチを作成する必要がある可能性があります。

セキュリティ、障害分離、IP アドレス重複の問題回避のために、これらのアプリケーションとテナントは互いに分離させる必要があります。仮想エンドポイントと物理エンドポイントは論理セグメントに接続し、データセンター ネットワーク内の物理的な位置に関係なく、接続を確立できます。これは、ネットワーク インフラストラクチャを、NSX-T のネットワーク仮想化による論理ネットワークから（つまり、基盤ネットワークをオーバーレイ ネットワークから）切り離すことで実現します。

分散論理ルーター

NSX-T の分散論理ルーターは、North-South 接続を提供するため、テナントからパブリック ネットワークへのアクセスが可能です。また、同じテナント内の異なるネットワーク間の East-West 接続も提供します。East-West 接続では、論理ルーターはホストのカーネル全体で分散されます。

NSX-T では、2 階層の分散論理ルーター トポロジを作成できます。上位の分散論理ルーターが Tier-0、下位の分散論理ルーターが Tier-1 です。この構成では、プロバイダ管理者とテナント管理者の両者が、それぞれのサービスとポリシーを完全に制御できます。管理者が Tier-0 のルーティングとサービスを制御および設定し、テナント管理者が Tier-1 を制御および設定します。Tier-0 の north 側の端は物理ネットワークとのインターフェイスになり、ここでダイナ

ミック ルーティング プロトコルを設定して、物理ルーターとルーティング情報を交換できます。Tier-0 の south 側の端は複数の Tier-1 ルーティング レイヤーと接続し、これらからルーティング情報を受け取ります。リソースの使用率を最適化するため、Tier-0 レイヤーは物理ネットワークから受け取るルートを手で Tier-1 にプッシュしませんが、デフォルト情報は提供します。

Tier-1 ルーティング レイヤーの south バウンドは、テナント管理者によって定義された論理スイッチと接続し、その論理スイッチとの間の 1 ホップルーティング機能を提供します。Tier-1 に接続されたサブネットに物理ネットワークからアクセスするには、Tier-0 レイヤー方向のルート再配分を有効にする必要があります。ただし、Tier-1 レイヤーと Tier-0 レイヤーの間に標準的なルーティング プロトコル (OSPF、BGP など) はなく、すべてのルートが NSX-T の制御プレーンを経由します。2 階層のルーティング トポロジは必須ではなく、プロバイダとテナントを分離する必要がない場合は 1 階層のトポロジを作成できます。この場合、論理スイッチは Tier-0 レイヤーに直接接続し、Tier-1 レイヤーはありません。

分散論理ルーターは 2 つのオプションで構成されます。1 つの分散ルーター (DR) と、1 つまたは複数のサービス ルーター (SR) です。

DR は、この分散論理ルーターに接続している仮想マシンのハイパーバイザーに加え、分散論理ルーターがバインドされている Edge ノードにまたがります。機能的には、DR は、この分散論理ルーターに接続している論理スイッチまたは分散論理ルーター、あるいはその両方の間で 1 ホップの分散ルーティングを担います。SR は、ステートフル NAT など、現在は分散式で実装されていないサービスの提供を担います。

分散論理ルーターには DR が必ずあり、次のいずれかの条件を満たす場合は SR があります。

- 分散論理ルーターが Tier-0 ルーターの場合。ステートフル サービスが設定されていない場合を含む。
- 分散論理ルーターが、Tier-0 ルーターにリンクされた Tier-1 ルーターであり、分散型の実装がないサービス (NAT、LB、DHCP など) が設定されている場合。

NSX-T の管理プレーン (MP) が、サービス ルーターを分散ルーターに接続する構成の自動作成を担います。MP は、中継論理スイッチを作成し、VNI を割り当ててから、各 SR と DR にポートを作成し、これらを中継論理スイッチに接続して、SR と DR に IP アドレスを割り当てます。

用語の説明

このドキュメントとユーザー インターフェイスで使用されている NSX-T の一般的な用語について説明します。

コンピュート マネージャ	コンピュート マネージャは、ホストや仮想マシンなどのリソースを管理するアプリケーションです。例：vCenter Server。
制御プレーン	管理プレーンからの設定に基づいてランタイム状態を算出します。制御プレーンは、データ プレーン要素からもたらされたトポロジ情報を伝達し、ステートレス設定をフォワーディング エンジンにプッシュします。
データ プレーン	制御プレーンが設定したテーブルに基づいて、パケットのステートレスな転送または変換を行います。データ プレーンはトポロジ情報を制御プレーンに報告し、パケット レベルの統計情報を保持します。

外部ネットワーク	NSX-T の管理対象ではない物理ネットワークまたは VLAN です。NSX Edge を通じて、論理ネットワークまたはオーバーレイ ネットワークを外部ネットワークにリンクできます。例として、お客様のデータセンター内の物理ネットワークや、物理環境内の VLAN などが挙げられます。
ファブリック ノード	NSX-T の管理プレーンに登録され、NSX-T モジュールがインストールされているホストです。ハイパーバイザー ホストまたは NSX Edge を NSX-T のオーバーレイの一部にするためには、NSX-T のファブリックに追加する必要があります。
論理ポート出力	仮想マシンまたは論理ネットワークから送信される送信ネットワーク トラフィックは、トラフィックが仮想ネットワークから出てデータセンターに入るため、出力方向と呼ばれます。
論理ポート入力	データセンターから出て仮想マシンに入る受信ネットワーク トラフィックは入力方向のトラフィックと呼ばれます。
論理ルーター	NSX-T のルーティング エンティティです。
論理ルーター ポート	論理スイッチ ポート、または物理ネットワークへのアップリンク ポートを関連付けることができる論理ネットワーク ポートです。
論理スイッチ	<p>仮想マシン インターフェイスとゲートウェイ インターフェイスに仮想レイヤー 2 スイッチングを提供するエンティティです。論理スイッチは、物理レイヤー 2 スイッチに対応する論理スイッチをテナント ネットワークの管理者に提供し、管理者が複数の仮想マシンを共通のプロードキャスト ドメインに接続できるようにします。論理スイッチは、物理ハイパーバイザー インフラストラクチャから独立した、多数のハイパーバイザーにまたがる論理エンティティであり、物理的な配置場所仮想マシンを接続します。</p> <p>マルチテナントのクラウドでは、各レイヤー 2 セグメントを相互に分離した状態で、多数の論理スイッチを同じハイパーバイザー ハードウェアに並べて配置できます。論理スイッチは論理ルーターを使用して接続でき、論理ルーターは外部物理ネットワークに接続したアップリンク ポートを提供できます。</p>
論理スイッチ ポート	仮想マシン ネットワーク インターフェイスまたは論理ルーター インターフェイスへの接続を確立する、論理スイッチの接続ポイントです。論理スイッチ ポートは、適用されているスイッチング プロファイル、ポートの状態、リンクのステータスをレポートします。
管理プレーン	システムへの単一の API エントリ ポイントで、ユーザー設定の維持とユーザー クエリの処理、システム内の管理プレーン、制御プレーン、データ プレーンのすべてのノードの処理を行います。管理プレーンは、ユーザー設定のクエリ、変更、維持を行います。
NSX Controller クラスタ	可用性に優れた仮想アプライアンスのクラスタとして展開され、NSX-T アーキテクチャ全体において、プログラムによる仮想ネットワークの展開を担います。
NSX Edge クラスタ	高可用性の監視にプロトコルと同じ設定を持つ NSX Edge ノード アプライアンスの集合。

NSX Edge ノード

IP ルーティングと IP サービスの機能に処理能力を提供することを機能的目標とするコンポーネント。

NSX 管理対象の分散仮想スイッチまたは KVM Open vSwitch

ハイパーバイザー上で実行され、トラフィック転送を行うソフトウェアです。NSX 管理対象の分散仮想スイッチ（以前ホストスイッチと呼ばれた N-VDS）または OVS は、テナント ネットワーク管理者には表示されずに、各論理スイッチが依存する基本の転送サービスを提供します。ネットワークの仮想化を実現するには、ネットワーク コントローラがハイパーバイザー仮想スイッチにネットワーク フロー テーブルを設定する必要があります。このフロー テーブルは、テナントの管理者が論理スイッチを作成および設定するときに定義した論理ブロードキャスト ドメインを形成します。

各論理ブロードキャスト ドメインは、トンネル カプセル化メカニズム Geneve を使用して、仮想マシン間のトラフィックと、仮想マシンと論理ルーター間のトラフィックをトンネリングすることで実装されます。ネットワーク コントローラが、データセンター全体を把握しており、仮想マシンの作成、移動、削除に伴ってハイパーバイザー仮想スイッチのフロー テーブルが更新されることを確認します。

N-VDS には標準と拡張データパスの 2 つのモードがあります。拡張データパスの N-VDS には、NFV (Network Functions Virtualization) ワークロードをサポートするパフォーマンス機能があります。

NSX Manager

API サービス、管理プレーン、エージェント サービスをホストするノードです。

NSX-T 統合アプライアンス

NSX-T 統合アプライアンスは、NSX-T インストール パッケージに含まれているアプライアンスです。NSX Manager、Policy Manager、または Cloud Service Manager のロールでアプライアンスを展開できます。現在、アプライアンスが一度にサポートできるロール数は 1 つのみです。

Open vSwitch (OVS)

XenServer、Xen、KVM、およびその他の Linux ベースのハイパーバイザーで仮想スイッチとして機能するオープン ソース ソフトウェア スイッチです。

オーバーレイ 論理ネットワーク

仮想マシンで認識されるトポロジが物理ネットワークのトポロジから切り離されるように、レイヤー 3 内のレイヤー 2 を使用して実装された論理ネットワークです。

物理インターフェイス (pNIC)

ハイパーバイザーがインストールされている物理サーバ上のネットワーク インターフェイスです。

Tier-0 論理ルーター

プロバイダ論理ルーターは、物理ネットワークとの Tier-0 論理ルーター インターフェイスとも呼ばれます。Tier-0 論理ルーターは最上位のルーターであり、サービス ルーターのアクティブ/アクティブ クラスタまたはアクティブ/スタンバイ クラスタとして実現できます。論理ルーターは BGP を実行し、物理ルーターとピアリングされます。アクティブ/スタンバイ モードでは、論理ルーターがステートフル サービスを提供することもできます。

Tier-1 論理ルーター

Tier-1 論理ルーターは、2 番目の論理ルーターです。North バウンド接続用に 1 台の Tier-0 論理ルーターと接続し、South バウンド接続用に 1 つ以上のオーバーレイ ネットワークと接続します。Tier-1 論理ルーターには、ステートフル サービスを提供するサービス ルーターのアクティブ/スタンバイ クラスタを使用できます。

トランスポート ゾーン	論理スイッチの最大範囲を定義するトランスポート ノードの集合。トランスポート ゾーンは、プロビジョニングされた一連のハイパーバイザーと、これらのハイパーバイザー上の仮想マシンを接続する論理スイッチを表します。
トランスポート ノード	NSX-T オーバーレイまたは NSX-T VLAN ネットワークに参加できるノード。KVM ホストの場合は、N-VDS を事前に設定できます。また、NSX Manager で設定を実行することも可能です。ESXi ホストの場合は、常に NSX Manager で N-VDS が設定されます。
アップリンク プロファイル	ハイパーバイザー ホストから NSX-T 論理スイッチまたは NSX Edge ノードからトップオブブラック スイッチへのリンクのポリシーを定義します。アップリンク プロファイルでは、チーミング ポリシー、アクティブ/スタンバイ リンク、トランスポート VLAN ID、MTU 設定などを定義します。
仮想マシン インターフェイス (vNIC)	仮想ゲスト OS と標準の vSwitch または vSphere Distributed Switch 間の接続を提供する、仮想マシンのネットワーク インターフェイスです。vNIC は論理ポートに接続できます。vNIC は、固有の ID (UUID) で識別できます。
仮想トンネルエンドポイント	ハイパーバイザー ホストを NSX-T のオーバーレイに加えることができます。NSX-T のオーバーレイは、パケット内にフレームをカプセル化し、基盤となるトランスポート ネットワーク上でパケットを送信することで、レイヤー 2 ネットワークを既存のレイヤー 3 ネットワーク ファブリック上に展開します。基盤となるトランスポート ネットワークは、別のレイヤー 2 ネットワークである場合と、レイヤー 3 の境界をまたぐ場合があります。VTEP は、カプセル化とカプセル化解除が行われる接続ポイントです。

インストールの準備

NSX-T をインストールする前に、導入環境の準備が完了していることを確認します。

この章には、次のトピックが含まれています。

- システム要件
- ポートとプロトコル
- NSX-T インストールの高水準のタスク

システム要件

NSX-T には、ハードウェア リソースとソフトウェア バージョンに固有の要件があります。

ハイパーバイザーの要件

ハイパーバイザー	バージョン	CPU コア	メモリ
vSphere	サポート対象の vSphere バージョン	4	16 GB
RHEL KVM	7.5 および 7.4	4	16 GB
Ubuntu KVM	16.04.2 LTS	4	16 GB
CentOS	7.4	4	16 GB

NSX-T でサポートされるのは、RHEL 7.5 および CentOS 7.4 のホストの準備のみです。RHEL 7.5 および CentOS 7.4 に NSX-T アプライアンスを展開することはできません。

ESXi の場合、NSX-T はホスト プロファイルおよび Auto Deploy 機能をサポートしません。



警告: Red Hat Enterprise Linux (RHEL) で **yum update** コマンドを実行すると、カーネルのバージョンがアップデートされ、NSX-T との互換性が失われることがあります。**yum update** を実行する場合は、カーネルの自動更新を無効にします。また、**yum install** を実行した後、NSX-T が該当のカーネルのバージョンをサポートしていることを確認します。

NSX Manager のリソース要件

シン仮想ディスクのサイズは 3.1 GB、シック仮想ディスクのサイズは 200 GB です。

アプライアンス	メモリ	vCPU	ストレージ	仮想マシンのハードウェアバージョン
NSX Manager の小規模な仮想マシン	8 GB	2	200 GB	10 以降
NSX Manager の中規模の仮想マシン	16 GB	4	200 GB	10 以降
NSX Manager の中規模/大規模の仮想マシン	24 GB	6	200 GB	10 以降
NSX Manager の大規模な仮想マシン	32 GB	8	200 GB	10 以降
NSX Manager 特大の仮想マシン	48 GB	12	200 GB	10 以降

注: NSX Manager の小規模な仮想マシンは、ラボおよび POC（事前検証）展開で使用する必要があります。

NSX Policy Manager と Cloud Service Manager には NSX Manager のリソース要件が適用されます。

NSX Controller のリソース要件

アプライアンス	メモリ	vCPU	ディスク容量	展開タイプ
NSX Controller の小規模な仮想マシン	8 GB	2	120 GB	ラボおよび POC（事前検証）展開
NSX Controller の中規模の仮想マシン	16 GB	4	120 GB	中規模の展開で推奨
NSX Controller の大規模な仮想マシン	32 GB	8	120 GB	大規模の展開で必須

注: 3 つの NSX Controller を展開して、高可用性を確保し、NSX-T 制御プレーンの停止を回避します。

NSX-T 制御プレーンに影響を与える、単一の物理ハイパーバイザー ホストに障害が発生するのを回避するために、各 NSX Controller クラスタは、3 台の独立した物理ハイパーバイザー ホストに配置する必要があります。『NSX-T リファレンス デザイン』ガイドを参照してください。

本番環境のワークロードのないラボや事前検証 (POC) 環境の場合は、リソースを節約するために単一の NSX Controller を配置することができます。

vSphere OVF 展開ユーザー インターフェイスからは小規模および大規模な仮想マシン フォーム ファクタを展開のみできます。

NSX Edge 仮想マシンのリソース要件

展開規模	メモリ	vCPU	ディスク容量	仮想マシンのハードウェア バージョン
小規模	4 GB	2	120 GB	10 以降 (vSphere 5.5 以降)
中規模	8 GB	4	120 GB	10 以降 (vSphere 5.5 以降)
大規模	16 GB	8	120 GB	10 以降 (vSphere 5.5 以降)

注: NSX Manager および NSX Edge の場合、小規模のアプライアンスは POC（事前検証）環境向けです。中規模のアプライアンスは標準的な本番環境に最適で、最大 64 のハイパーバイザーをサポートできます。大規模のアプライアンスは、64 を超えるハイパーバイザーを使用する大規模環境用です。

注: VMXNET 3 vNIC は、NSX Edge 仮想マシンでのみサポートされます。

NSX Edge 仮想マシンとベア メタル NSX Edge CPU の要件

ハードウェア	タイプ
CPU	<ul style="list-style-type: none"> ■ Xeon 56xx (Westmere-EP) ■ Xeon E7-xxxx (Westmere-EX 以降の世代の CPU) ■ Xeon E5-xxxx (Sandy Bridge 以降の世代の CPU)

ベア メタル NSX Edge 固有の NIC 要件

NIC タイプ	説明	PCI デバイス ID
Intel X520/Intel 82599	IXGBE_DEV_ID_82599_KX4	0x10F7
	IXGBE_DEV_ID_82599_KX4_MEZZ	0x1514
	IXGBE_DEV_ID_82599_KR	0x1517
	IXGBE_DEV_ID_82599_COMBO_BACKPL	0x10F8
	ANE	0x000C
	IXGBE_SUBDEV_ID_82599_KX4_KR_MEZ	0x10F9
	Z	0x10FB
	IXGBE_DEV_ID_82599_CX4	0x11A9
	IXGBE_DEV_ID_82599_SFP	0x1F72
	IXGBE_SUBDEV_ID_82599_SFP	0x17D0
	IXGBE_SUBDEV_ID_82599_RNDC	0x0470
	IXGBE_SUBDEV_ID_82599_560FLR	0x1507
	IXGBE_SUBDEV_ID_82599_ECNA_DP	0x154D
	IXGBE_DEV_ID_82599_SFP_EM	0x154A
	IXGBE_DEV_ID_82599_SFP_SF2	0x1558
	IXGBE_DEV_ID_82599_SFP_SF_QP	0x1557
	IXGBE_DEV_ID_82599_QSFP_SF_QP	0x10FC
	IXGBE_DEV_ID_82599EN_SFP	0x151C
	IXGBE_DEV_ID_82599_XAUI_LOM	
	IXGBE_DEV_ID_82599_T3_LOM	
Intel X540	IXGBE_DEV_ID_X540T	0x1528
	IXGBE_DEV_ID_X540T1	0x1560
Intel X550	IXGBE_DEV_ID_X550T	0x1563
	IXGBE_DEV_ID_X550T1	0x15D1
Intel X710	I40E_DEV_ID_SFP_X710	0x1572
	I40E_DEV_ID_KX_C	0x1581
	I40E_DEV_ID_10G_BASE_T	0x1586
Intel XL710	I40E_DEV_ID_KX_B	0x1580
	I40E_DEV_ID_QSFP_A	0x1583
	I40E_DEV_ID_QSFP_B	0x1584
	I40E_DEV_ID_QSFP_C	0x1585

ベア メタル NSX Edge のメモリ、CPU、ディスクの要件

メモリ	CPU コア	ディスク容量
32 GB	8	200 GB

NSX Manager のブラウザのサポート

ブラウザ	Windows 10	Windows 8.1	Ubuntu 14.04	Mac OS X 10.11 および 10.12
Internet Explorer 11	○	○		
Firefox 55			○	○
Chrome 60	○	○		○
Safari 10				○
Microsoft Edge 40	○			

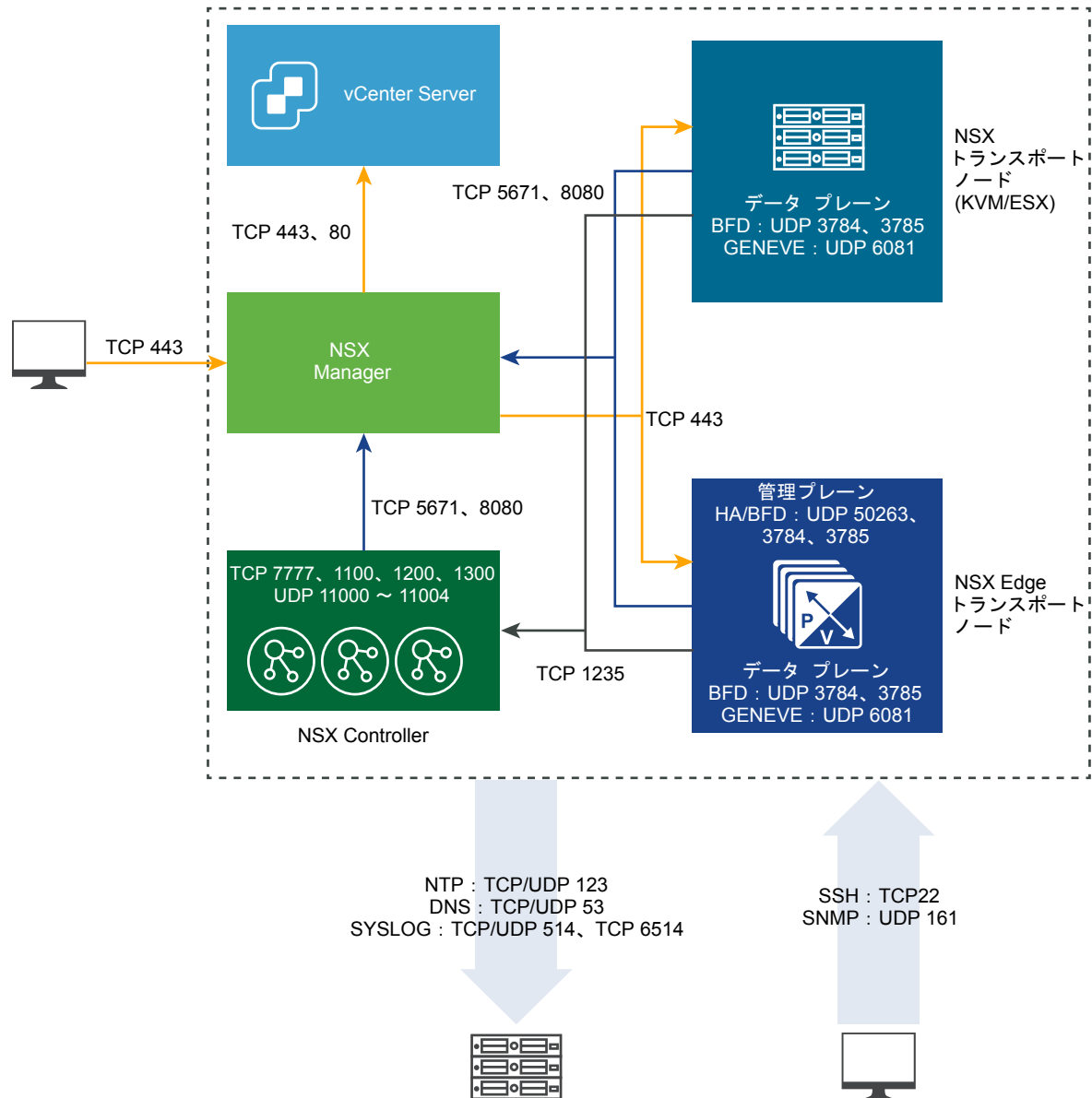
注: Internet Explorer 11 の互換モードはサポートされていません。

サポートされるブラウザの最小解像度は、1280 × 800 ピクセルです。

ポートとプロトコル

ポートとプロトコルにより、NSX-T でノード間の通信パスが使用可能になります。このパスはセキュリティで保護され、認証が行われます。また、資格情報の保管場所を使って相互認証が行われます。

図 2-1. NSX-T のポートとプロトコル



デフォルトでは、すべての証明書が自己署名の証明書です。ノースパウンドのユーザー インターフェイス、API 証明書、プライベート キーは、CA 署名付きの証明書で置き換えることができます。

ループバックまたは UNIX ドメインのソケットを経由して通信する内部デーモンがあります。

- KVM : MPA、netcpa、nsx-agent、OVS
- ESX : netcpa、ESX-DP (カーネル内)

RMQ ユーザー データベース (db) では、パスワードが不可逆性のハッシュ関数でハッシュされます。h (p1) はパスワード p1 のハッシュです。

CCP 中央制御プレーン

LCP ローカル制御プレーン

MP 管理プレーン

MPA 管理プレーン エージェント

注: NSX-T ノードにアクセスするには、これらのノードで SSH を有効にする必要があります。

NSX Manager が使用する TCP および UDP ポート

NSX Manager は、特定の TCP および UDP ポートを使用して、他のコンポーネントおよび製品と通信します。ファイアウォールで、これらのポートを開く必要があります。

API 呼び出しまたは CLI コマンドを使用して、ファイルを転送するためのカスタム ポート（デフォルトは 22）および Syslog データをエクスポートするためのカスタム ポート（デフォルトは 514 および 6514）を指定することができます。その場合は、ファイアウォールを適切に設定する必要があります。

表 2-1. NSX Manager が使用する TCP および UDP ポート

送信元	宛先	ポート	プロトコル	説明
管理クライアント	NSX Manager	22	TCP	SSH（デフォルトでは無効）
NTP サーバ	NSX Manager	123	UDP	NTP
管理クライアント	NSX Manager	443	TCP	NSX API サーバ
SNMP サーバ	NSX Manager	161	UDP	SNMP
NSX Controller、NSX Edge ノード、 トランスポート ノード	NSX Manager	8080	TCP	インストールとアップグレードの HTTP リポジトリ
NSX Controller、NSX Edge ノード、 トランスポート ノード	NSX Manager	5671	TCP	NSX メッセージング
NSX Manager	管理 SCP サーバ	22	TCP	SSH（サポートバンドル、バックアップなどのアップロード）
NSX Manager	DNS サーバ	53	TCP	DNS
NSX Manager	DNS サーバ	53	UDP	DNS
NSX Manager	NTP サーバ	123	UDP	NTP
NSX Manager	SNMP サーバ	161、 162	TCP	SNMP
NSX Manager	SNMP サーバ	161、 162	UDP	SNMP
NSX Manager	Syslog サーバ	514	TCP	Syslog
NSX Manager	Syslog サーバ	514	UDP	Syslog
NSX Manager	Syslog サーバ	6514	TCP	Syslog
NSX Manager	Syslog サーバ	6514	UDP	Syslog
NSX Manager	LogInsight サーバ	9000	TCP	Log Insight エージェント

表 2-1. NSX Manager が使用する TCP および UDP ポート (続き)

送信元	宛先	ポート	プロトコル	説明
NSX Manager	Traceroute の宛先	334 34 - 335 23	UDP	Traceroute
NSX Manager	vCenter Server	80	TCP	設定されている場合、NSX Manager からコンピュートマネージャ (vCenter Server) への通信。
NSX Manager	vCenter Server	443	TCP	設定されている場合、NSX Manager からコンピュートマネージャ (vCenter Server) への通信。

NSX Controller が使用する TCP および UDP ポート

NSX Controller は、特定の TCP および UDP ポートを使用して、他のコンポーネントおよび製品と通信します。ファイアウォールで、これらのポートを開く必要があります。

API 呼び出しまたは CLI コマンドを使用して、ファイルを転送するためのカスタム ポート (デフォルトは 22) および Syslog データをエクスポートするためのカスタム ポート (デフォルトは 514 および 6514) を指定することができます。その場合は、ファイアウォールを適切に設定する必要があります。

表 2-2. NSX Controller が使用する TCP および UDP ポート

送信元	宛先	ポート	プロトコル	説明
管理クライアント	NSX Controller	22	TCP	SSH (デフォルトでは無効)
DNS サーバ	NSX Controller	53	UDP	DNS
NTP サーバ	NSX Controller	123	UDP	NTP
SNMP サーバ	NSX Controller	161	UDP	SNMP
NSX Controller	NSX Controller	1100	TCP	Zookeeper クォーラム
NSX Controller	NSX Controller	1200	TCP	Zookeeper リーダー選出
NSX Controller	NSX Controller	1300	TCP	Zookeeper サーバ
NSX Edge ノード、トランスポート ノード	NSX Controller	1235	TCP	CCP-netcpa 通信
NSX Controller	NSX Controller	7777	TCP	Moot RPC
NSX Controller	NSX Controller	11000 - 11004	UDP	他のクラスタ ノードへのトンネル。クラスタのノード数が 5 台より多い場合は、さらに多くのポートを開く必要があります。
Traceroute の宛先	NSX Controller	33434 - 33523	UDP	Traceroute
NSX Controller	SSH の宛先	22	TCP	SSH (デフォルトでは無効)
NSX Controller	DNS サーバ	53	UDP	DNS
NSX Controller	DNS サーバ	53	TCP	DNS
NSX Controller	任意	80	TCP	HTTP
NSX Controller	NTP サーバ	123	UDP	NTP

表 2-2. NSX Controller が使用する TCP および UDP ポート (続き)

送信元	宛先	ポート	プロトコル	説明
NSX Controller	NSX Manager	5671	TCP	NSX メッセージング
NSX Controller	LogInsight サーバ	9000	TCP	Log Insight エージェント
NSX Controller	NSX Controller	11000 - 11004	TCP	他のクラスタ ノードへのトンネル。クラスタのノード数が 5 台より多い場合は、さらに多くのポートを開く必要があります。
NSX Controller	NSX Manager	8080	TCP	NSX のアップグレード
NSX Controller	Traceroute の宛先	33434 - 33523	UDP	Traceroute
NSX Controller	Syslog サーバ	514	UDP	Syslog
NSX Controller	Syslog サーバ	514	TCP	Syslog
NSX Controller	Syslog サーバ	6514	TCP	Syslog

NSX Edge が使用する TCP および UDP ポート

NSX Edge は、特定の TCP および UDP ポートを使用して、他のコンポーネントおよび製品と通信します。ファイアウォールで、これらのポートを開く必要があります。

API 呼び出しまたは CLI コマンドを使用して、ファイルを転送するためのカスタム ポート（デフォルトは 22）および Syslog データをエクスポートするためのカスタム ポート（デフォルトは 514 および 6514）を指定することができます。その場合は、ファイアウォールを適切に設定する必要があります。

表 2-3. NSX Edge が使用する TCP および UDP ポート

送信元	宛先	ポート	プロトコル	説明
管理クライアント	NSX Edge	22	TCP	SSH（デフォルトでは無効）
NTP サーバ	NSX Edge	123	UDP	NTP
SNMP サーバ	NSX Edge	161	UDP	SNMP
テナントのワークロード	NSX Edge	67、68	UDP	ホストするテナント ワークロードへの DHCP サービス
任意	NSX Edge	1167	TCP	DHCP バックエンド
NSX Edge ノード、トランスポート ノード	NSX Edge	3784、3785	UDP	データ内のトランスポート ノード TEP IP アドレス間の BFD。
NSX Agent	NSX Edge	5555	TCP	NSX クラウド：インスタンス上のエージェントが NSX Cloud Gateway と通信します。
NSX Edge ノード	NSX Edge	6666	TCP	NSX クラウド：NSX Edge ローカル通信。
NSX Edge ノード	NSX Manager	8080	TCP	NAPI、NSX-T のアップグレード
NSX Edge ノード	NSX Edge	2480	TCP	Nestdb
NSX Edge ノード	管理 SCP または SSH サーバ	22	TCP	SSH
NSX Edge ノード	DNS サーバ	53	UDP	DNS
NSX Edge ノード	任意	80	TCP	HTTP

表 2-3. NSX Edge が使用する TCP および UDP ポート (続き)

送信元	宛先	ポート	プロトコル	説明
NSX Edge ノード	NTP サーバ	123	UDP	NTP
NSX Edge ノード	SNMP サーバ	161、 162	UDP	SNMP
NSX Edge ノード	SNMP サーバ	161、 162	TCP	SNMP
NSX Edge ノード	外部ルーター	179	TCP	T-0 論理ルーターと外部の物理ルーター間の BGP。
NSX Edge ノード	NSX Manager	443	TCP	HTTPS
NSX Edge ノード	Syslog サーバ	514	TCP	Syslog
NSX Edge ノード	Syslog サーバ	514	UDP	Syslog
NSX Edge ノード	任意	1167	TCP	DHCP バックエンド
NSX Edge ノード	NSX Controller	1235	TCP	netcpa
NSX Edge ノード	任意	3000 - 9000	TCP	メタデータ プロキシ
NSX Edge ノード	NSX Manager	5671	TCP	NSX メッセージング
NSX Edge ノード	Syslog サーバ	6514	TCP	TLS を介した Syslog
NSX Edge ノード	Traceroute の宛先	33434 - 33523	UDP	Traceroute
NSX Edge ノード	NSX Edge ノード	50263	UDP	高可用性

vSphere ESXi および KVM ホストによって使用される TCP および UDP ポート

トランスポート ノードとして vSphere ESXi ホストおよび KVM ホストを使用する場合には、特定の TCP ポートおよび UDP ポートが必要です。

表 2-4. vSphere ESXi および KVM ホストによって使用される TCP および UDP ポート

送信元	宛先	ポート	プロトコル	説明
NSX Manager	vSphere ESXi ホスト	443	TCP	管理とプロビジョニング接続
NSX Manager	KVM ホスト	443	TCP	管理とプロビジョニング接続
vSphere ESXi ホスト	NSX Manager	567 1	TCP	NSX Manager との AMQP 通信チャンネル
vSphere ESXi ホスト	NSX Controller	123 5	TCP	制御プレーン - LCP と CCP 間の通信
KVM ホスト	NSX Manager	567 1	TCP	NSX Manager との AMQP 通信チャンネル
KVM ホスト	NSX Controller	123 5	TCP	制御プレーン - LCP と CCP 間の通信
vSphere ESXi ホスト	NSX Manager	808 0	TCP	HTTP リポジトリのインストールおよびアップグレード

表 2-4. vSphere ESXi および KVM ホストによって使用される TCP および UDP ポート (続き)

送信元	宛先	ポート	プロトコル	説明
KVM ホスト	NSX Manager	8080	TCP	HTTP リポジトリのインストールおよびアップグレード
GENEVE Termination End Point (TEP)	GENEVE Termination End Point (TEP)	6081	UDP	トランスポート ネットワーク
NSX-T トランスポート ノード	NSX-T トランスポート ノード	3784、 3785	UDP	TEP インターフェイスを使用するデータベースにおける TEPS 間の BFD セッション

NSX-T インストールの高水準のタスク

チェックリストを使用してインストールの進行状況を追跡します。

推奨される手順は次のとおりです。

- 1 NSX Manager をインストールします。[章 4 「NSX Manager のインストール」](#) を参照してください。
- 2 NSX Controller をインストールします。[章 5 「NSX Controller のインストールとクラスタリング」](#) を参照してください。
- 3 NSX Controller を管理プレーンに追加します。[「NSX Manager への NSX Controller の追加」](#) を参照してください。
- 4 マスターの NSX Controller を作成し、コントロール クラスタを初期化します。[「コントロール クラスタの初期化によるコントロール クラスタ マスターの作成」](#) を参照してください。
- 5 NSX Controller をコントロール クラスタに追加します。[「クラスタ マスターを使用した NSX Controller の追加」](#) を参照してください。

ハイパーバイザー ホストを追加した後、NSX Manager は NSX-T モジュールをインストールします。

注: NSX-T モジュールをインストールすると、ハイパーバイザー ホストに証明書が作成されます。

- 6 ハイパーバイザー ホストを管理プレーンに追加します。[「ハイパーバイザー ホストの管理プレーンへの追加」](#) を参照してください。
ホストは、ホスト証明書を管理プレーンに送信します。
- 7 NSX Edge をインストールします。[章 6 「NSX Edge のインストール」](#) を参照してください。
- 8 NSX Edge を管理プレーンに追加します。[「NSX Edge の管理プレーンへの追加」](#) を参照してください。
- 9 トランスポート ゾーンとトランスポート ノードを作成します。[章 8 「トランスポート ゾーンとトランスポート ノード」](#) を参照してください。

各ホストで仮想スイッチが作成されます。管理プレーンからホストの証明書が制御プレーンに送信され、管理プレーンによって制御プレーンの情報がホストにプッシュされます。各ホストは、証明書を提示して SSL 経由で制御プレーンに接続します。制御プレーンは、管理プレーンから提供されたホスト証明書に基づいて証明書を検証します。検証が正常に完了すると、コントローラが接続を許可します。

標準的なインストール順序は、次のとおりです。

- 1 NSX Manager を最初にインストールします。
- 2 NSX Controller をインストールし、管理プレーンに参加させることができます。
- 3 ハイパーバイザー ホストを管理プレーンに追加する前に、NSX-T モジュールをハイパーバイザー ホストにインストールできます。また、ユーザー インターフェイスの [ファブリック] > [ホスト] > [追加] を使用して両方の処理を同時に行うこともできます。
- 4 NSX Controller、NSX Edge、NSX-T モジュールをインストールしたホストは、いつでも管理プレーンに追加できます。

インストール後

ホストがトランスポート ノードの場合、NSX Manager のユーザー インターフェイスまたは API を使用して、トランスポート ゾーン、論理スイッチ、論理ルーター、その他のネットワーク コンポーネントをいつでも作成できます。NSX Controller、NSX Edge、ホストを管理プレーンに追加するときに、NSX-T の論理エンティティと設定状態が自動的に NSX Controller、NSX Edge、ホストにプッシュされます。

詳細については、『NSX-T 管理ガイド』を参照してください。

KVM の使用

NSX-T は、2 種類の方法で KVM をサポートします。KVM は、1) ホスト トランスポート ノードとして、2) NSX Manager と NSX Controller のホストとして使用できます。

表 3-1. KVM のサポート対象バージョン

要件	説明
サポート対象のプラットフォーム	<ul style="list-style-type: none">■ RHEL 7.4■ Ubuntu 16.04.2 LTS

この章には、次のトピックが含まれています。

- [KVM のセットアップ](#)
- [KVM CLI を使用したゲスト仮想マシンの管理](#)

KVM のセットアップ

トランスポート ノードとして、または NSX Manager や NSX Controller ゲスト仮想マシンのホストとして KVM を使用する予定で、KVM のセットアップが完了していない場合、次の手順を実行します。

注: Geneve カプセル化プロトコルは UDP ポート 6081 を使用します。KVM ホストのファイアウォールで、このポートへのアクセスを許可する必要があります。

手順

- 1 (Red Hat のみ) `/etc/yum.conf` ファイルを開きます。
- 2 「`exclude`」行を検索します。
- 3 「`"kernel* redhat-release*"`」行を追加し、サポートされていない RHEL のアップグレードが回避されるように yum を設定します。

```
exclude=[existing list] kernel* redhat-release*
```

サポートされている RHEL のバージョンは 7.4 です。

4 KVM とブリッジユーティリティをインストールします。

Linux ディストリビューション	コマンド
Ubuntu	<code>apt-get install -y qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils virtinst virt-manager virt-viewer libguestfs-tools</code>
RHEL	<pre> yum groupinstall "Virtualization Hypervisor" yum groupinstall "Virtualization Client" yum groupinstall "Virtualization Platform" yum groupinstall "Virtualization Tools" </pre>

5 ハードウェアが仮想化に対応しているか確認します。

```
cat /proc/cpuinfo | egrep "vmx|svm"
```

コマンド出力に「vmx」が含まれることを確認します。

6 KVM モジュールがインストールされていることを確認します。

Linux ディストリビューション	コマンド
Ubuntu	<pre> kvm-ok INFO: /dev/kvm exists KVM acceleration can be used </pre>
RHEL	<pre> lsmod grep kvm kvm_intel 53484 6 kvm 316506 1 kvm_intel </pre>

- 7 KVM を NSX Manager または NSX Controller のホストとして使用する場合は、ブリッジ ネットワーク、管理 インターフェイス、および NIC インターフェイスを準備します。

次の例では、1 つめのイーサネット インターフェイス (eth0 または ens32) が Linux マシン自体への接続に使用されます。このインターフェイスでは、導入環境に応じて DHCP または固定 IP アドレス設定を使用します。NSX-T ホストにアップリンク インターフェイスを割り当てる前に、これらのアップリンクによって使用される インターフェイス スクリプトを必ず設定しておきます。システムにスクリプトのインターフェイス ファイルがない場合は、ホスト トランスポート ノードを作成できません。

注: インターフェイス名は環境によって異なる場合があります。

Linux ディストリ ビューション	ネットワーク設定
Ubuntu	<p>/etc/network/interfaces を次のように編集します。</p> <pre> auto lo iface lo inet loopback auto eth0 iface eth0 inet manual auto br0 iface br0 inet static address 192.168.110.51 netmask 255.255.255.0 network 192.168.110.0 broadcast 192.168.110.255 gateway 192.168.110.1 dns-nameservers 192.168.3.45 dns-search example.com bridge_ports eth0 bridge_stp off bridge_fd 0 bridge_maxwait 0 </pre> <p>ブリッジにネットワークを定義する xml ファイルを作成します。たとえば、次の行で /tmp/bridge.xml を作成します。</p> <pre> <network> <name>bridge</name> <forward mode='bridge' /> <bridge name='br0' /> </network> </pre> <p>次のコマンドでブリッジ ネットワークを定義し、開始します。</p> <pre> virsh net-define bridge.xml virsh net-start bridge virsh net-autostart bridge </pre>

Linux ディストリ
ビューション

ネットワーク設定

次のコマンドでブリッジ ネットワークのステータスを確認できます。

```
virsh net-list --all
```

Name	State	Autostart	Persistent
bridge	active	yes	yes
default	active	yes	yes

RHEL

/etc/sysconfig/network-scripts/ifcfg-**<management_interface>** を次のように編集します。

```
DEVICE="ens32"
TYPE="Ethernet"
NAME="ens32"
UUID="<<UUID>>"
BOOTPROTO="none"
HWADDR="<<HWADDR>>"
ONBOOT="yes"
NM_CONTROLLED="no"
BRIDGE="br0"
```

/etc/sysconfig/network-scripts/ifcfg-eth1 を次のように編集します。

```
DEVICE="eth1"
TYPE="Ethernet"
NAME="eth1"
UUID="<<UUID>>"
BOOTPROTO="none"
HWADDR="<<HWADDR>>"
ONBOOT="yes"
NM_CONTROLLED="no"
```

/etc/sysconfig/network-scripts/ifcfg-eth2 を次のように編集します。

```
DEVICE="eth2"
TYPE="Ethernet"
NAME="eth2"
UUID="<<UUID>>"
BOOTPROTO="none"
HWADDR="<<HWADDR>>"
ONBOOT="yes"
NM_CONTROLLED="no"
```

/etc/sysconfig/network-scripts/ifcfg-br0 を次のように編集します。

```
DEVICE="br0"
BOOTPROTO="dhcp"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Bridge"
```

8 KVM をトランスポート ノードとして使用する場合、ネットワーク ブリッジを準備します。

次の例では、1 つめのイーサネット インターフェイス (eth0 または ens32) が Linux マシン自体への接続に使用されます。このインターフェイスでは、導入環境に応じて DHCP または固定 IP アドレス設定を使用します。

注: インターフェイス名は環境によって異なる場合があります。

Linux ディストリ ビューション	ネットワーク設定
Ubuntu	<p>/etc/network/interfaces を次のように編集します。</p> <pre> auto lo iface lo inet loopback auto eth0 iface eth0 inet manual auto eth1 iface eth1 inet manual auto br0 iface br0 inet dhcp bridge_ports eth0 </pre>
RHEL	<p>/etc/sysconfig/network-scripts/ifcfg-ens32 ファイルを次のように編集します。</p> <pre> DEVICE="ens32" TYPE="Ethernet" NAME="ens32" UUID="<something>" BOOTPROTO="none" HWADDR="<something>" ONBOOT="yes" NM_CONTROLLED="no" BRIDGE="br0" </pre> <p>/etc/sysconfig/network-scripts/ifcfg-ens33 ファイルを次のように編集します。</p> <pre> DEVICE="ens33" TYPE="Ethernet" NAME="ens33" UUID="<something>" BOOTPROTO="none" HWADDR="<something>" ONBOOT="yes" NM_CONTROLLED="no" </pre> <p>/etc/sysconfig/network-scripts/ifcfg-br0 を次のように編集します。</p> <pre> DEVICE="br0" BOOTPROTO="dhcp" NM_CONTROLLED="no" ONBOOT="yes" TYPE="Bridge" </pre>

重要: Ubuntu の場合、すべてのネットワーク設定を `/etc/network/interfaces` で指定する必要があります。`/etc/network/ifcfg-eth1` など、ネットワーク設定ファイルは個別に作成しないでください。トランスポート ノードの作成に失敗する可能性があります。

この手順の後、KVM ホストをトランスポート ノードとして設定すると、ブリッジ インターフェイス「`nsx-vtep0.0`」が作成されます。Ubuntu では、`/etc/network/interfaces` に次のようなエントリが記述されます。

```
iface nsx-vtep0.0 inet static
pre-up ip addr flush dev nsx-vtep0.0
address <IP_pool_address>
netmask <subnet_mask>
mtu 1600
down ifconfig nsx-vtep0.0 down
up ifconfig nsx-vtep0.0 up
```

RHEL では、ホスト NSX Agent (nsxa) によって「`ifcfg-nsx-vtep0.0`」という設定ファイルが作成され、次のようなエントリが記述されます。

```
DEVICE=nsx-vtep0.0
BOOTPROTO=static
NETMASK=<IP address>
IPADDR=<subnet mask>
MTU=1600
ONBOOT=yes
USERCTL=no
NM_CONTROLLED=no
```

- 9 ネットワークの変更を有効にするには、ネットワーク サービス `systemctl restart network` を再起動するか、Linux サーバを再起動します。

KVM CLI を使用したゲスト仮想マシンの管理

NSX Manager と NSX Controller は、KVM 仮想マシンとしてインストールできます。また、KVM を NSX-T トランスポート ノードのハイパーバイザーとして使用することもできます。

KVM のゲスト仮想マシンの管理は、このガイドの対象範囲外です。ここでは簡単な KVM CLI コマンドを紹介します。

KVM CLI でゲスト仮想マシンを管理するには、`virsh` コマンドを使用します。一般的な `virsh` コマンドをいくつか示します。詳細については、KVM のドキュメントを参照してください。

```
# List running
virsh list

# List all
virsh list --all

# Control instances
virsh start <instance>
virsh shutdown <instance>
virsh destroy <instance>
virsh undefine <instance>
```

```
virsh suspend <instance>
virsh resume <instance>

# Access an instance's CLI
virsh console <instance>
```

Linux CLI では、**ifconfig** コマンドが、ゲスト仮想マシン用に作成されたインターフェイスを表す vnetX インターフェイスを表示します。ゲスト仮想マシンを追加すると、vnetX インターフェイスが追加されます。

```
ifconfig
...

[vnet0]      Link encap:Ethernet  HWaddr fe:54:00:b0:a0:6d
            inet6 addr: fe80::fc54:ff:feb0:a06d/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:13183 errors:0 dropped:0 overruns:0 frame:0
            TX packets:181524 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:500
            RX bytes:4984832 (4.9 MB)  TX bytes:29498709 (29.4 MB)
```

NSX Manager のインストール

NSX Manager には、論理スイッチ、論理ルーター、ファイアウォールなどの NSX-T コンポーネントを作成、設定、監視するためのグラフィカル ユーザー インターフェイス (GUI) と REST API があります。

NSX Manager はシステム ビューを提供するものであり、NSX-T の管理コンポーネントです。

NSX Manager の 1 つのインスタンスのみをインストールすることができます。ESXi ホスト上で NSX Manager を展開すると、vSphere 高可用性 (HA) 機能を利用して NSX Manager の可用性を確保できます。

表 4-1. NSX Manager の展開、プラットフォームおよびインストール要件

要件	説明
サポートされる展開方法	<ul style="list-style-type: none"> ■ OVA/OVF ■ QCOW2
サポート対象のプラットフォーム	<p>「システム要件」を参照してください。</p> <p>ESXi では、共有ストレージに NSX Manager アプライアンスをインストールすることが推奨されます。vSphere 高可用性を使用する場合、元のホストに障害が発生したときに仮想マシンを別のホストで再起動できるように、共有ストレージが必要になります。</p>
IP アドレス	NSX Manager には固定 IP アドレスが必要です。インストール後に IP アドレスを変更することはできません。
NSX-T アプライアンスのパスワード	<ul style="list-style-type: none"> ■ 8 文字以上 ■ 1 文字以上の小文字 ■ 1 文字以上の大文字 ■ 1 文字以上の数字 ■ 1 文字以上の特殊文字 ■ 5 文字以上の異なる文字 ■ 辞書に登録されている単語が使われていない ■ パリンドローム (回文) になっていない
ホスト名	NSX Manager をインストールするときに、アンダースコアなどの無効な文字を含まないホスト名を指定します。ホスト名に無効な文字が含まれていると、展開後にホスト名が nsx-manager に設定されます。ホスト名の制限の詳細については、 https://tools.ietf.org/html/rfc952 および https://tools.ietf.org/html/rfc1123 を参照してください。
VMware Tools	ESXi で実行される NSX Manager 仮想マシンには、VMware Tools がインストールされています。VMware Tools を削除またはアップグレードしないでください。

表 4-1. NSX Manager の展開、プラットフォームおよびインストール要件 (続き)

要件	説明
システム	<ul style="list-style-type: none"> ■ システム要件を満たしていることを確認します。「システム要件」を参照してください。 ■ 必要なポートが開いていることを確認します。「ポートとプロトコル」を参照してください。 ■ まだ作成していない場合は、宛先の仮想マシン ポート グループ ネットワークを作成します。管理仮想マシン ネットワークに NSX-T アプライアンスを配置することをお勧めします。 <p>複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。</p> <ul style="list-style-type: none"> ■ IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。
OVF の権限	<p>ESXi ホストに OVF テンプレートを展開するために必要な権限があることを確認します。</p> <p>OVF テンプレートを展開できる管理ツール (vCenter Server、vSphere Client など) が必要です。手動で設定できるようにするには、OVF 展開ツールで設定オプションがサポートされている必要があります。</p> <p>OVF ツールは、4.0 以降のバージョンを使用する必要があります。</p>
クライアント プラグイン	クライアント統合プラグインがインストールされている必要があります。

注: NSX Manager のフレッシュ インストールや再起動時、また初回のログイン時にプロンプトで **admin** のパスワードを変更した後は、NSX Manager の起動に数分かかる場合があります。

NSX Manager のインストール シナリオ

重要: vSphere Web Client または コマンド ラインのいずれかを使用して OVA または OVF ファイルから NSX Manager をインストールすると、仮想マシンがパワーオン状態になるまで、ユーザー名、パスワード、IP アドレスなどの OVA/OVF プロパティ 値が検証されません。

- **admin** または **audit** ユーザーのユーザー名を指定する場合には、一意の名前を使用する必要があります。同じ名前を指定すると、名前が無視され、デフォルトの名前 (**admin** または **audit**) が使用されます。
- **admin** ユーザーのパスワード要件を満たしていない場合には、**admin** ユーザーとして SSH またはコンソール経由で NSX Manager にログインする必要があります。プロンプトが表示され、パスワードの変更が指示されます。
- **audit** ユーザーのパスワードが要件を満たしていない場合、ユーザー アカウントは無効になります。アカウントを有効にするには、**admin** ユーザーとして SSH またはコンソール経由で NSX Manager にログインし、**set user audit** コマンドを実行して **audit** ユーザーのパスワードを設定します (現在のパスワードは空の文字列です)。
- **root** ユーザーのパスワード要件を満たしていない場合には、**root** として SSH またはコンソール経由で NSX Manager にログインする必要があります。ログイン パスワードは **vmware** です。プロンプトが表示され、パスワードの変更が指示されます。

- この製品で操作を実行する場合、**root** ユーザーの認証情報を使用しないでください。このアクセスは、VMware サポート チームから要求された場合にのみ使用してください。**root** ユーザーの認証情報を使用してデーモンまたはアプリケーションをインストールすると、サポート契約が無効になります。

注: アプライアンス上のコア サービスは、要件を満たすパスワードが設定されるまで起動しません。

NSX Manager を OVA ファイルから展開した後は、仮想マシンをパワーオフして vCenter Server から OVA 設定を変更し、仮想マシンの IP アドレス設定を変更することはできません。

この章には、次のトピックが含まれています。

- [NSX Manager および利用可能なアプライアンスのインストール](#)
- [コマンドライン OVF ツールを使用した ESXi への NSX Manager のインストール](#)
- [KVM への NSX Manager のインストール](#)
- [新しく作成された NSX Manager にログインします。](#)

NSX Manager および利用可能なアプライアンスのインストール

vSphere Web Client を使用して NSX Manager、NSX Policy Manager、または Cloud Service Manager を仮想アプライアンスとして展開できます。

NSX Policy Manager は、ポリシーを管理できる仮想アプライアンスです。論理ポート、IP アドレス、仮想マシンなどの NSX-T コンポーネント向けのルールを指定するようにポリシーを設定できます。NSX Policy Manager ルールを使用すると、上位レベルの使用状況とリソース アクセスのルールを設定し、詳細な内容を指定せずに適用することができます。

Cloud Service Manager は、NSX-T コンポーネントを使用する仮想アプライアンスで、パブリック クラウドへのコンポーネントの組み込みを行います。

注: vSphere Client ではなく、vSphere Web Client を使用することが推奨されます。環境内に vCenter Server がない場合は、**ovftool** を使用して NSX Manager を展開します。「[\[コマンドライン OVF ツールを使用した ESXi への NSX Manager のインストール\]](#)」を参照してください。

手順

- 1 NSX-T 統合アプライアンスの OVA ファイルまたは OVF ファイルの場所を確認します。
ダウンロード URL をコピーするか、OVA ファイルをコンピュータにダウンロードします。
- 2 vSphere Web Client で [OVF テンプレートの展開] ウィザードを起動し、.ova ファイルを指定します。
- 3 NSX Manager の名前を入力し、フォルダまたはデータセンターを選択します。
ここに入力する名前がインベントリに表示されます。
選択したフォルダを使用して、NSX Manager に権限が適用されます。
- 4 NSX Manager の仮想アプライアンス ファイルを格納するデータストアを選択します。

- 5 vCenter Server にインストールする場合は、NSX Manager アプライアンスを展開するホストまたはクラスタを選択します。

通常は、ネットワーク管理機能を備えたクラスタに NSX Manager を配置します。

- 6 NSX Manager のポート グループまたはインストール先ネットワークを選択します。
- 7 NSX Manager のパスワードと IP アドレスを指定します。
- 8 [nsx-manager] ロールをそのまま使用します。
 - ドロップダウン メニューから [nsx-policy-manager] ロールを選択して、NSX Policy Manager アプライアンスをインストールします。
 - ドロップダウン メニューから [nsx-cloud-service-manager] ロールを選択して、NSX Cloud アプライアンスをインストールします。

注: [nsx-manager nsx-cloud-service-manager (multi-role)] ロールはサポートされていません。

- 9 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。
メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。
- 10 NSX-T コンポーネントのコンソールを開いて、ブート プロセスを追跡します。
- 11 NSX-T コンポーネントが起動した後、admin として CLI にログインし、**get interface eth0** コマンドを実行して、IP アドレスが想定どおりに適用されていることを確認します。

```
nsx-component> get interface eth0
Interface: eth0
Address: 192.168.110.25/24
MAC address: 00:50:56:86:7b:1b
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

- 12 NSX-T コンポーネントに必要な接続があることを確認します。

次のタスクを実行できることを確認します。

- 別のマシンから NSX-T コンポーネントに ping を実行します。
- NSX-T コンポーネントは、デフォルト ゲートウェイに ping を実行できます。
- NSX-T コンポーネントは、管理インターフェイスを使用して、NSX-T コンポーネントと同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX-T コンポーネントは、DNS サーバと NTP サーバに ping を実行できます。
- SSH を有効にした場合は、SSH を使用して NSX-T コンポーネントに接続できることを確認します。

接続が確立されていない場合は、仮想アプライアンスのネットワーク アダプタが適切なネットワークまたは VLAN に配置されていることを確認します。

次のステップ

サポート対象のブラウザを使用して、NSX Manager GUI に接続します。

URL は `https://<IP address of NSX Manager>` です。たとえば、`https://10.16.176.10` などです。

注: HTTPS を使用する必要があります。HTTP はサポートされていません。

コマンドライン OVF ツールを使用した ESXi への NSX Manager のインストール

NSX Manager のインストールを自動的に行うか、CLI で行う場合は、コマンドライン ユーティリティの VMware OVF ツールを使用します。

デフォルトでは、`nsx_isSSHEnabled` と `nsx_allowSSHRootLogin` はいずれもセキュリティ上の理由より無効になっています。無効になっている場合、NSX Manager のコマンドラインへの SSH 接続やログインはできません。`nsx_isSSHEnabled` を有効にして、`nsx_allowSSHRootLogin` を有効にしなかった場合、NSX Manager に SSH で接続することはできますが、`root` でログインすることはできません。

前提条件

- システム要件を満たしていることを確認します。[「システム要件」](#)を参照してください。
- 必要なポートが開いていることを確認します。[「ポートとプロトコル」](#)を参照してください。
- まだ作成していない場合は、宛先の仮想マシン ポート グループ ネットワークを作成します。管理仮想マシン ネットワークに NSX-T アプライアンスを配置することをお勧めします。

複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。

- IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。

手順

- スタンドアロン ホストの場合、適切なパラメータを指定して **ovftool** コマンドを実行します。

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-manager
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--net="management"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_role=nsx-manager
--prop:nsx_ip_0=192.168.110.75
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
```

```

--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-manager
nsx-<component>.ova
vi://root:<password>@192.168.110.51

```

```

Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://root:<password>@192.168.110.51
Deploying to VI: vi://root:<password>@192.168.110.51
Transfer Completed
Powering on VM: nsx-manager
Task Completed
Completed successfully

```

- vCenter Server で管理されているホストの場合、適切なパラメータを指定して **ovftool** コマンドを実行します。次はその例です。

```

C:\Users\Administrator\Downloads>ovftool
--name=nsx-manager
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--network="management"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_role=nsx-manager
--prop:nsx_ip_0=192.168.110.75
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>

```

```
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-manager
nsx-<component>.ova
vi://administrator@vsphere.local:<password>@192.168.110.24/?ip=192.168.110.51
```

```
Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://administrator@vsphere.local@192.168.110.24:443/
Deploying to VI: vi://administrator@vsphere.local@192.168.110.24:443/
Transfer Completed
Powering on VM: nsx-manager
Task Completed
Completed successfully
```

- (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。

- NSX-T コンポーネントのコンソールを開いて、ブート プロセスを追跡します。
- NSX-T コンポーネントが起動した後、admin として CLI にログインし、**get interface eth0** コマンドを実行して、IP アドレスが想定どおりに適用されていることを確認します。

```
nsx-component> get interface eth0
Interface: eth0
  Address: 192.168.110.25/24
  MAC address: 00:50:56:86:7b:1b
  MTU: 1500
  Default gateway: 192.168.110.1
  Broadcast address: 192.168.110.255
  ...
```

- NSX-T コンポーネントに必要な接続があることを確認します。

次のタスクを実行できることを確認します。

- 別のマシンから NSX-T コンポーネントに ping を実行します。
- NSX-T コンポーネントは、デフォルト ゲートウェイに ping を実行できます。
- NSX-T コンポーネントは、管理インターフェイスを使用して、NSX-T コンポーネントと同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX-T コンポーネントは、DNS サーバと NTP サーバに ping を実行できます。
- SSH を有効にした場合は、SSH を使用して NSX-T コンポーネントに接続できることを確認します。

接続が確立されていない場合は、仮想アプライアンスのネットワーク アダプタが適切なネットワークまたは VLAN に配置されていることを確認します。

次のステップ

サポート対象のブラウザを使用して、NSX Manager GUI に接続します。

URL は `https://<IP address of NSX Manager>` です。たとえば、`https://10.16.176.10` などです。

注: HTTPS を使用する必要があります。HTTP はサポートされていません。

KVM への NSX Manager のインストール

NSX Manager は、KVM ホストに仮想アプライアンスとしてインストールできます。

QCOW2 のインストール手順では、`guestfish` という Linux のコマンドライン ツールを使用して、仮想マシンの設定を QCOW2 ファイルに書き込みます。

前提条件

- KVM が構成されていること。「[「KVM のセットアップ」](#)」を参照してください。
- QCOW2 イメージを KVM ホストに展開する権限。
- インストール後にログインできるように、`guestinfo` のパスワードがパスワードの強度の要件に準拠していることを確認します。「[章 4 「NSX Manager のインストール」](#)」を参照してください。

手順

- 1 NSX Manager QCOW2 イメージをダウンロードし、SCP によりファイル転送をするか、`sync` を使用して、NSX Manager を実行する KVM マシンにコピーします。
- 2 (Ubuntu のみ) 現在ログインしているユーザーを `libvirtd` ユーザーとして追加します。

```
adduser $USER libvirtd
```

- 3 QCOW2 イメージを保存したディレクトリに `guestinfo` というファイル（ファイル拡張子なし）を作成し、NSX Manager 仮想マシンのプロパティを入力します。

次はその例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="nsx_role" oe:value="nsx-manager"/>
    <Property oe:key="nsx_allowSSHRootLogin" oe:value="True"/>
    <Property oe:key="nsx_cli_passwd_0" oe:value="<password>"/>
    <Property oe:key="nsx_dns1_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_domain_0" oe:value="corp.local"/>
    <Property oe:key="nsx_gateway_0" oe:value="192.168.110.1"/>
    <Property oe:key="nsx_hostname" oe:value="nsx-manager1"/>
    <Property oe:key="nsx_ip_0" oe:value="192.168.110.19"/>
    <Property oe:key="nsx_isSSHEnabled" oe:value="True"/>
  </PropertySection>
</Environment>
```

```
<Property oe:key="nsx_netmask_0" oe:value="255.255.255.0"/>
<Property oe:key="nsx_ntp_0" oe:value="192.168.110.10"/>
<Property oe:key="nsx_passwd_0" oe:value="<password>"/>
</PropertySection>
</Environment>
```

この例では、`nsx_isSSHEnabled` と `nsx_allowSSHRootLogin` がいずれも有効になっています。無効になっている場合、NSX Manager のコマンドラインへの SSH 接続やログインはできません。`nsx_isSSHEnabled` を有効にして、`nsx_allowSSHRootLogin` を有効にしなかった場合、NSX Manager に SSH で接続することはできますが、`root` でログインすることはできません。

- 4 `guestfish` を使用して **guestinfo** ファイルを QCOW2 イメージに書き込みます。

guestinfo の情報を QCOW2 イメージに書き込んだ後、情報を上書きすることはできません。

```
sudo guestfish --rw -i -a nsx-manager1-build.qcow2 upload guestinfo /config/guestinfo
```

- 5 `virt-install` コマンドで QCOW2 イメージを展開します。

```
user@ubuntu1604:/var/lib/libvirt/images$ sudo virt-install --import --name nsx-manager1
--ram 16348 --vcpus 4 --network=bridge:br0,model=e1000 --disk
path=/var/lib/libvirt/images/nsx-manager-1.1.0.0.4446302.qcow2,format=qcow2 --
nographics

Starting install...
Creating domain...      |    0 B    00:01
Connected to domain nsx-manager1
Escape character is ^]

nsx-manager1 login:
```

NSX Manager が起動したら、NSX Manager コンソールが表示されます。

- 6 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。

- 7 NSX-T コンポーネントのコンソールを開いて、ブート プロセスを追跡します。
- 8 NSX-T コンポーネントが起動した後、`admin` として CLI にログインし、**get interface eth0** コマンドを実行して、IP アドレスが想定どおりに適用されていることを確認します。

```
nsx-component> get interface eth0
Interface: eth0
Address: 192.168.110.25/24
MAC address: 00:50:56:86:7b:1b
```

```
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

9 NSX-T コンポーネントに必要な接続があることを確認します。

次のタスクを実行できることを確認します。

- 別のマシンから NSX-T コンポーネントに ping を実行します。
- NSX-T コンポーネントは、デフォルト ゲートウェイに ping を実行できます。
- NSX-T コンポーネントは、管理インターフェイスを使用して、NSX-T コンポーネントと同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX-T コンポーネントは、DNS サーバと NTP サーバに ping を実行できます。
- SSH を有効にした場合は、SSH を使用して NSX-T コンポーネントに接続できることを確認します。

接続が確立されていない場合は、仮想アプライアンスのネットワーク アダプタが適切なネットワークまたは VLAN に配置されていることを確認します。

10 KVM コンソールを終了します。

control-]

次のステップ

サポート対象のブラウザを使用して、NSX Manager GUI に接続します。

URL は `https://<IP address of NSX Manager>` です。たとえば、`https://10.16.176.10` などです。

注: HTTPS を使用する必要があります。HTTP はサポートされていません。

新しく作成された NSX Manager にログインします。

NSX Manager をインストールした後は、ユーザー インターフェイスを使用して、その他のインストール タスクを実行できます。

NSX Manager をインストールしたら、NSX-T のカスタマー エクスペリエンス向上プログラム (CEIP) に参加できます。プログラムへの参加または参加を中止する方法については、『NSX-T 管理ガイド』の NSX のカスタマー エクスペリエンス向上プログラムのセクションを参照してください。

前提条件

NSX Manager がインストールされていることを確認します。

手順

- 1 ブラウザから、NSX Manager(`https://<nsx-manager-ip-address>`)に管理者権限でログインします。
エンド ユーザー使用許諾契約書 (EULA) が表示されます。
- 2 EULA の一番下までスクロールし、条件に同意します。

- 3 VMware のカスタマー エクスペリエンス向上プログラム (CEIP) に参加するかどうかを選択します。
- 4 [保存] をクリックします。

NSX Controller のインストールとクラスタリング

5

NSX Controller は、NSX-T の論理スイッチングおよびルーティング機能の制御プレーンとして機能する高度な分散状態管理システムです。

NSX Controller は、ネットワーク内のすべての論理スイッチの集中管理ポイントとして機能し、すべてのホスト、論理スイッチ、および論理ルーターの情報を管理します。NSX Controller は、パケット転送を行うデバイスを制御します。これらの転送デバイスを仮想スイッチといいます。

NSX が管理する分散仮想スイッチ（以前はホストスイッチと呼ばれていた N-VDS）や Open vSwitch (OVS) などの仮想スイッチは、ESXi あるいは KVM などのその他のハイパーバイザーに常駐しています。

本番環境では、NSX 制御プレーンの停止を回避するために、3 台のホストをメンバーに持つ NSX Controller クラスタが必要です。1 台の物理ハイパーバイザー ホストで発生した障害による NSX 制御プレーンへの影響を回避するには、各コントローラを個別のハイパーバイザー ホスト（合計 3 台の物理ハイパーバイザー ホスト）に配置する必要があります。本番環境のワークロードを処理しないラボや事前検証 (POC) 環境の場合は、リソースを節約するために単一のコントローラで実行することもできます。

表 5-1. NSX Controller の展開、プラットフォームおよびインストール要件

要件	説明
サポートされる展開方法	<ul style="list-style-type: none">■ OVA/OVF■ QCOW2 <p><u>注:</u> PXE ブートによる展開方法はサポートされていません。</p>
サポート対象のプラットフォーム	<ul style="list-style-type: none">■ サポート対象の vSphere バージョン■ RHEL 7.4■ Ubuntu 16.04.2 LTS <p>NSX Controller は、ESXi（仮想マシン）と KVM でサポートされます。</p> <p><u>注:</u> PXE ブートによる展開方法はサポートされていません。</p>
IP アドレス	<p>NSX Controller には固定 IP アドレスが必要です。インストール後に IP アドレスを変更することはできません。</p> <p>IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。</p>

表 5-1. NSX Controller の展開、プラットフォームおよびインストール要件 (続き)

要件	説明
NSX-T アプライアンスのパスワード	<ul style="list-style-type: none"> ■ 8 文字以上 ■ 1 文字以上の小文字 ■ 1 文字以上の大文字 ■ 1 文字以上の数字 ■ 1 文字以上の特殊文字 ■ 5 文字以上の異なる文字 ■ 辞書に登録されている単語が使われていない ■ パリンドローム (回文) になっていない
ホスト名	NSX Controller をインストールするときに、アンダースコアなどの無効な文字を含まないホスト名を指定します。ホスト名に無効な文字が含まれていると、展開後にホスト名が localhost に設定されます。ホスト名の制限の詳細については、 https://tools.ietf.org/html/rfc952 および https://tools.ietf.org/html/rfc1123 を参照してください。
VMware Tools	ESXi で実行される NSX Controller 仮想マシンには、VMware Tools がインストールされています。VMware Tools を削除またはアップグレードしないでください。
システム	システム要件を満たしていることを確認します。「 システム要件 」を参照してください。
ポート	必要なポートが開いていることを確認します。「 ポートとプロトコル 」を参照してください。

NSX Controller のインストール シナリオ

重要: vSphere Web Client またはコマンド ラインのいずれかを使用して OVA または OVF ファイルから NSX Controller をインストールすると、仮想マシンがパワーオン状態になるまで、ユーザー名、パスワード、IP アドレスなどの OVA/OVF プロパティ値が検証されません。

- **admin** または **audit** ユーザーのユーザー名を指定する場合には、一意の名前を使用する必要があります。同じ名前を指定すると、名前が無視され、デフォルトの名前 (**admin** または **audit**) が使用されます。
- **admin** ユーザーのパスワード要件を満たしていない場合には、**admin** ユーザーとして SSH またはコンソール経由で NSX Controller にログインする必要があります。プロンプトが表示され、パスワードの変更が指示されます。
- **audit** ユーザーのパスワードが要件を満たしていない場合、ユーザー アカウントは無効になります。アカウントを有効にするには、**admin** ユーザーとして SSH またはコンソール経由で NSX Controller にログインし、**set user audit** コマンドを実行して **audit** ユーザーのパスワードを設定します (現在のパスワードは空の文字列です)。
- **root** ユーザーのパスワード要件を満たしていない場合には、**root** として SSH またはコンソール経由で NSX Controller にログインする必要があります。ログイン パスワードは **vmware** です。プロンプトが表示され、パスワードの変更が指示されます。

- この製品で操作を実行する場合、**root** ユーザーの認証情報を使用しないでください。このアクセスは、VMware サポート チームから要求された場合にのみ使用してください。**root** ユーザーの認証情報を使用してデーモンまたはアプリケーションをインストールすると、サポート契約が無効になります。

注:

- root 権限を使用してデーモンまたはアプリケーションをインストールしないでください。root 権限を使用してデーモンまたはアプリケーションをインストールすると、サポート契約が無効になることがあります。root 権限は、VMware のサポート チームから要求された場合にのみ使用します。
- 要件を満たすパスワードが設定されるまで、コア サービスはアプライアンスで起動しません。
NSX Controller を OVA ファイルから展開した後は、仮想マシンをパワーオフにして vCenter Server から OVA の設定を変更し、仮想マシンの IP 設定を変更することはできません。

この章には、次のトピックが含まれています。

- [NSX Manager からのコントローラとクラスタの自動インストール](#)
- [グラフィカル ユーザー インターフェイス \(GUI\) を使用した ESXi への NSX Controller のインストール](#)
- [コマンドライン OVF ツールを使用した ESXi への NSX Controller のインストール](#)
- [KVM への NSX Controller のインストール](#)
- [NSX Manager への NSX Controller の追加](#)
- [コントロール クラスタの初期化によるコントロール クラスタ マスターの作成](#)
- [クラスタ マスターを使用した NSX Controller の追加](#)

NSX Manager からのコントローラとクラスタの自動インストール

NSX Manager を設定して vCenter Server が管理する vSphere ESXi ホストにコントローラを自動的に展開し、これらのホストを vCenter Server が管理する vSphere ESXi ホストのコントローラ クラスタに自動的に追加するようにします。自動展開は、NSX Manager のユーザー インターフェイスから、または API で有効にできます。

コントローラを手動で追加していた既存のクラスタを、コントローラを自動で展開するように設定できます。ただし、手動で追加したコントローラは、手動で削除する必要があります。

サポートされる使用方法

- 単一のノードで構成されるクラスタの作成
- 複数のノードで構成されるクラスタの作成
- 既存のクラスタへのノードの追加
- 機能クラスタからの自動展開されたコントローラの削除

NSX Manager のユーザー インターフェイスを使用したコントローラとクラスタの自動インストールの設定

vCenter Server によって管理されている vSphere ESXi ホストにコントローラを自動的に展開し、vCenter Server によって管理されている vSphere ESXi ホストのコントローラ クラスタに自動的に追加するように NSX Manager を設定します。

前提条件

- NSX Manager を展開します。
- vCenter Server および vSphere ESXi ホストを展開します。
- vSphere ESXi ホストを vCenter Server に登録します。
- vSphere ESXi ホストには、12 個の vCPU、48 GB の RAM、および 360 GB のストレージをサポートするために十分な CPU、メモリ、およびハード ディスク リソースが必要です。

手順

- 1 NSX Manager (<https://<<NSX Manager の IP アドレス>>>) にログインします。
- 2 NSX Manager ユーザー インターフェイスで、登録済みの vCenter Server がない場合は、[ファブリック] パネルに進み、[コンピュート マネージャ] をクリックしてコンピュート マネージャを追加します。
- 3 [システム] ページで、[コントローラの追加] をクリックします。
- 4 [共通属性] ページで、必要な値を入力します。
- 5 [コンピュート マネージャ] を選択します。
- 6 (オプション) SSH を有効にします。
- 7 (オプション) root アクセスを有効にします。
- 8 (オプション) 既存のクラスタに新しいノードを追加する場合は、[既存のクラスタに追加] を有効にします。
- 9 クラスタの初期化および編成に必要な共有シークレット キーを入力して確認します。

注: このクラスタに追加されたすべてのコントローラ ノードが同じ共有シークレット キーを使用していることを確認します。

- 10 コントローラ認証情報を入力します。
- 11 [次へ] をクリックします。
- 12 [コントローラ] ページで、[コントローラの追加] をクリックします。
- 13 コントローラ ノードの有効なホスト名または完全修飾ドメイン名を入力します。
- 14 クラスタを選択します。
- 15 (オプション) リソース プールを選択します。リソース プールは、コントローラ ノードを展開するためのコンピュート リソース プールのみを提供します。これには、特定のストレージ リソースを割り当てる必要があります。
- 16 (オプション) ホストを選択します。

- 17 データ ストアを選択します。
- 18 ホストがホスト内部のさまざまなコンポーネントと通信するために使用する管理インターフェイスを選択します。
- 19 ポートの詳細を含む固定 IP アドレス (<<IP アドレス>>/<<ポート番号>>) およびネット マスクを入力します。
- 20 複数のコントローラを追加できます。展開を開始する前に、[+] ボタンをクリックしてコントローラの詳細を入力します。
- 21 [終了] をクリックして、コントローラの自動展開を開始します。
コントローラを NSX Manager に登録してから、クラスタを作成するか、既存のクラスタへの追加を行います。
- 22 コントローラが NSX Manager に登録されているかどうかを確認します。
 - a NSX Manager コンソールにログインします。
 - b `# get management-cluster status` と入力します。
管理クラスタのステータスは、「STABLE」となっている必要があります。
 - c あるいは、NSX Manager ユーザー インターフェイスで NSX Manager の接続が「稼動中」であることを確認します。
- 23 コントロール クラスタの状態を確認します。
 - a コントローラ CLI コンソールにログインします。
 - b `# get control-cluster status` と入力します。
コントローラ クラスタのステータスは、「STABLE」となっている必要があります。
 - c あるいは、NSX Manager のユーザー インターフェイスでクラスタの接続が「稼動中」であることを確認します。

次のステップ

API を使用して、コントローラとクラスタの自動インストールを設定します。[「API を使用したコントローラとクラスタの自動インストールの設定」](#)を参照してください。

API を使用したコントローラとクラスタの自動インストールの設定

API を使用して、vCenter Server によって管理されている vSphere ESXi ホストにコントローラを自動的に展開し、vCenter Server によって管理されている vSphere ESXi ホストのコントローラ クラスタに自動的に追加するように NSX Manager を設定します。

手順

- 1 コントローラ クラスタの自動作成を開始する前に、POST API のペイロードとして必要となる vCenter Server ID、コンピュータ ID、ストレージ ID、およびネットワーク ID を取得する必要があります。
- 2 vCenter Server にログインします。
`https://<vCenterServer_IPAddress>/mob.`
- 3 [値] 列で、[コンテンツ] をクリックします。

- 4 [コンテンツのプロパティ] ページで、[値] 列に移動し、データセンターを検索して、グループ リンクをクリックします。
- 5 [グループのプロパティ] ページで、[値] 列に移動し、データセンター リンクをクリックします。
- 6 [データセンターのプロパティ] ページで、コントローラ クラスタの作成で使用するデータストア値およびネットワーク値をコピーします。
- 7 同じページで、[ホスト フォルダ] リンクをクリックします。
- 8 [グループのプロパティ] ページで、コントローラ クラスタの作成に使用するクラスタ値をコピーします。
- 9 vCenter Server ID を取得するには、NSX Manager のユーザー インターフェイスに移動し、[コンピュート マネージャ] ページから使用する ID をコピーします。
- 10 POST <https://<nsx-manager>/api/v1/cluster/nodes/deployments>

```

REQUEST
{
  "deployment_requests": [
    {
      "roles": ["CONTROLLER"],
      "user_settings": {
        "cli_password": "CLIp4$$w4rd",
        "root_password": "ROOTp4$$w4rd"
      },
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
        "management_network_id": "network-13",
        "hostname": "controller-0",
        "compute_id": "domain-s9",
        "storage_id": "datastore-12",
        "default_gateway_addresses": [
          "10.33.79.253"
        ],
        "management_port_subnets": [
          {
            "ip_addresses": [
              "10.33.79.64"
            ],
            "prefix_length": "22"
          }
        ]
      }
    },
    {
      "roles": ["CONTROLLER"],
      "user_settings": {
        "cli_password": "VMware$123",
        "root_password": "VMware$123"
      },
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",

```

```

        "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
        "management_network_id": "network-13",
        "hostname": "controller-1",
        "compute_id": "domain-s9",
        "storage_id": "datastore-12"
        "default_gateway_addresses": [
            "10.33.79.253"
        ],
        "management_port_subnets": [
            {
                "ip_addresses": [
                    "10.33.79.65"
                ],
                "prefix_length": "22"
            }
        ]
    }
},
    "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
        "management_network_id": "network-13",
        "hostname": "controller-0",
        "compute_id": "domain-s9",
        "storage_id": "datastore-12",
        "default_gateway_addresses": [
            "10.33.79.253"
        ],
        "management_port_subnets": [
            {
                "ip_addresses": [
                    "10.33.79.66"
                ],
                "prefix_length": "22"
            }
        ]
    }
},
    "clustering_config": {
        "clustering_type": "ControlClusteringConfig",
        "shared_secret": "123456",
        "join_to_existing_cluster": false
    }
}

```

Response

```

{
    "result_count": 2,
    "results": [
        {
            "user_settings": {
                "cli_password": "[redacted]",
                "root_password": "[redacted]",
            }
        }
    ]
}

```

```

    "cli_username": "admin"
  },
  "vm_id": "71f02260-644f-4482-aa9a-ab8570bb49a3",
  "roles": [
    "CONTROLLER"
  ],
  "deployment_config": {
    "placement_type": "VsphereClusterNodeVMDeploymentConfig",
    "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
    "management_network_id": "network-13",
    "default_gateway_addresses": [
      "10.33.79.253"
    ],
    "hostname": "controller-0",
    "compute_id": "domain-s9",
    "storage_id": "datastore-12",
    "management_port_subnets": [
      {
        "ip_addresses": [
          "10.33.79.64"
        ],
        "prefix_length": 22
      }
    ]
  },
  "form_factor": "SMALL"
},
{
  "user_settings": {
    "cli_password": "[redacted]",
    "root_password": "[redacted]",
    "cli_username": "admin"
  },
  "vm_id": "38029a2b-b9bc-467f-8138-aef784e802cc",
  "roles": [
    "CONTROLLER"
  ],
  "deployment_config": {
    "placement_type": "VsphereClusterNodeVMDeploymentConfig",
    "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
    "management_network_id": "network-13",
    "hostname": "controller-1",
    "compute_id": "domain-s9",
    "storage_id": "datastore-12"
  },
  "form_factor": "MEDIUM"
}
]
}

```

- 11 API 呼び出しを使用して展開のステータスを表示できます。GET <https://<nsx-manager>/api/v1/cluster/nodes/deployments>

```
{
  "result_count": 2,
  "results": [
    {
      "user_settings": {
        "cli_password": "[redacted]",
        "root_password": "[redacted]"
      },
      "vm_id": "12f563af-af9f-48f3-848e-e9257c8740b0",
      "roles": [
        "CONTROLLER"
      ],
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "15145422-47a1-4c55-81da-01d953151d1f",
        "management_network_id": "network-158",
        "hostname": "controller-0",
        "compute_id": "domain-c154",
        "storage_id": "datastore-157"
      },
      "form_factor": "SMALL",
    },
    {
      "user_settings": {
        "cli_password": "[redacted]",
        "root_password": "[redacted]"
      },
      "vm_id": "cc21854c-265b-42de-af5f-05448c00777a",
      "roles": [
        "CONTROLLER"
      ],
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "feb17651-49a7-4ce6-88b4-41d3f624e53b",
        "management_network_id": "network-158",
        "hostname": "controller-0",
        "compute_id": "domain-c154",
        "storage_id": "datastore-157"
      },
      "form_factor": "MEDIUM",
    }
  ]
}
```

次のステップ

クラスタを削除します。「[自動的に展開されたコントローラの削除](#)」を参照してください。

自動的に展開されたコントローラの削除

自動的に展開されたコントローラをクラスタから削除します。

手順

- 1 **https://<nsx-manager-ip>/** にログインします。**
- 2 [システム (System)] > [コンポーネント (Components)] の順にクリックします。
- 3 [コントローラ クラスタ] で、コントローラを特定します。
- 4 [設定 (Settings)] アイコンをクリックし、[削除 (Delete)] をクリックします。
- 5 [確認 (Confirm)] をクリックします。

NSX でコントローラがクラスタから切り離され、NSX Manager から登録解除され、パワーオフされた後、コントローラが削除されます。

グラフィカル ユーザー インターフェイス (GUI) を使用した ESXi への NSX Controller のインストール

NSX Controller を対話形式でインストールする場合は、ユーザー インターフェイス ベースの仮想マシン管理ツールを使用できます。たとえば、vSphere Client を vCenter Server に接続して使用します。

パスワードが要件を満たしていない場合でも、インストールは成功します。ただし、初回ログイン時にパスワードの変更を求められます。

重要: 要件を満たすパスワードが設定されるまで、コア サービスはアプライアンスで起動しません。

重要: NSX-T コンポーネント仮想マシンのインストールには VMware Tools が含まれます。NSX-T アプライアンスで VMware Tools を削除またはアップグレードすることはできません。

前提条件

- システム要件を満たしていることを確認します。[「システム要件」](#)を参照してください。
- 必要なポートが開いていることを確認します。[「ポートとプロトコル」](#)を参照してください。
- まだ作成していない場合は、宛先の仮想マシン ポート グループ ネットワークを作成します。管理仮想マシン ネットワークに NSX-T アプライアンスを配置することをお勧めします。

複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。
- IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。
- ESXi ホストに OVF テンプレートを展開するために必要な権限があることを確認します。

- ホスト名にアンダースコアが含まれていないことを確認します。そうでない場合は、ホスト名が <nsx-controller> に設定されます。
- OVF テンプレートを展開できる管理ツールが必要です (vCenter Server や vSphere Client など)。
手動で設定できるようにするには、OVF 展開ツールで設定オプションがサポートされている必要があります。
- クライアント統合プラグインがインストールされている必要があります。

手順

- 1 NSX Controller の OVA ファイルまたは OVF ファイルの場所を確認します。
ダウンロード URL をコピーするか、OVA ファイルをコンピュータにダウンロードします。
- 2 管理ツールで [OVF テンプレートの展開] ウィザードを起動し、.ova ファイルを指定します。
- 3 NSX Controller の名前を入力し、フォルダまたはデータセンターを選択します。
ここに入力する名前がインベントリに表示されます。
選択したフォルダを使用して、NSX Controller に権限が適用されます。
- 4 NSX Controller の仮想アプライアンス ファイルを格納するデータストアを選択します。
- 5 vCenter Server を使用している場合は、NSX Controller アプライアンスを展開するホストまたはクラスタを選択します。
通常は、ネットワーク管理機能を備えたクラスタに NSX Controller を配置します。
- 6 NSX Controller のポート グループまたはインストール先ネットワークを選択します。
- 7 NSX Controller のパスワードと IP アドレスを指定します。
- 8 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。
メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#) を参照してください。
- 9 NSX-T コンポーネントのコンソールを開いて、ブート プロセスを追跡します。
- 10 NSX-T コンポーネントが起動した後、admin として CLI にログインし、**get interface eth0** コマンドを実行して、IP アドレスが想定どおりに適用されていることを確認します。

```
nsx-component> get interface eth0
Interface: eth0
Address: 192.168.110.25/24
MAC address: 00:50:56:86:7b:1b
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

11 NSX-T コンポーネントに必要な接続があることを確認します。

次のタスクを実行できることを確認します。

- 別のマシンから NSX-T コンポーネントに ping を実行します。
- NSX-T コンポーネントは、デフォルト ゲートウェイに ping を実行できます。
- NSX-T コンポーネントは、管理インターフェイスを使用して、NSX-T コンポーネントと同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX-T コンポーネントは、DNS サーバと NTP サーバに ping を実行できます。
- SSH を有効にした場合は、SSH を使用して NSX-T コンポーネントに接続できることを確認します。

接続が確立されていない場合は、仮想アプライアンスのネットワーク アダプタが適切なネットワークまたは VLAN に配置されていることを確認します。

次のステップ

NSX Controller を管理プレーンに追加します。「[「NSX Manager への NSX Controller の追加」](#)」を参照してください。

コマンドライン OVF ツールを使用した ESXi への NSX Controller のインストール

NSX Controller のインストールを自動的に行う場合は、コマンドライン ユーティリティの VMware OVF Tool を使用します。

デフォルトでは、`nsx_isSSHEnabled` と `nsx_allowSSHRootLogin` はいずれもセキュリティ上の理由より無効になっています。無効になっている場合、NSX Controller のコマンドラインへの SSH 接続やログインはできません。`nsx_isSSHEnabled` を有効にして、`nsx_allowSSHRootLogin` を有効にしなかった場合、NSX Controller に SSH で接続することはできますが、`root` でログインすることはできません。

前提条件

- システム要件を満たしていることを確認します。「[システム要件](#)」を参照してください。
- 必要なポートが開いていることを確認します。「[ポートとプロトコル](#)」を参照してください。
- まだ作成していない場合は、宛先の仮想マシン ポート グループ ネットワークを作成します。管理仮想マシン ネットワークに NSX-T アプライアンスを配置することをお勧めします。
複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。
- IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。
- OVF Tool バージョン 4.0 以降。

手順

- スタンドアロン ホストの場合、適切なパラメータを指定して **ovftool** コマンドを実行します。

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-controller
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--network="management"
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.210
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_cli_audit_passwd_0=<password>
--prop:nsx_hostname=nsx-controller
<path/url to nsx component ova>
vi://root:<password>@192.168.110.51
```

- vCenter Server で管理されているホストの場合、適切なパラメータを指定して **ovftool** コマンドを実行します。

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-controller
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--network="management"
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.210
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
```

```
--prop:nsx_cli_audit_passwd_0=<password>
--prop:nsx_hostname=nsx-controller
<path/url to nsx component ova>
vi://administrator@vsphere.local:<vcenter_password>@192.168.110.24/?ip=192.168.110.51
```

- (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。

- NSX-T コンポーネントのコンソールを開いて、ブート プロセスを追跡します。
- NSX-T コンポーネントが起動した後、admin として CLI にログインし、**get interface eth0** コマンドを実行して、IP アドレスが想定どおりに適用されていることを確認します。

```
nsx-component> get interface eth0
Interface: eth0
Address: 192.168.110.25/24
MAC address: 00:50:56:86:7b:1b
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

- NSX-T コンポーネントに必要な接続があることを確認します。

次のタスクを実行できることを確認します。

- 別のマシンから NSX-T コンポーネントに ping を実行します。
- NSX-T コンポーネントは、デフォルト ゲートウェイに ping を実行できます。
- NSX-T コンポーネントは、管理インターフェイスを使用して、NSX-T コンポーネントと同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX-T コンポーネントは、DNS サーバと NTP サーバに ping を実行できます。
- SSH を有効にした場合は、SSH を使用して NSX-T コンポーネントに接続できることを確認します。

接続が確立されていない場合は、仮想アプライアンスのネットワーク アダプタが適切なネットワークまたは VLAN に配置されていることを確認します。

次のステップ

NSX Controller を管理プレーンに追加します。[「**NSX Manager への NSX Controller の追加**」](#)を参照してください。

KVM への NSX Controller のインストール

NSX Controller は、ネットワーク内のすべての論理スイッチの集中管理ポイントとして機能し、すべてのホスト、論理スイッチ、および論理ルーターの情報を管理します。

QCOW2 のインストール手順では、guestfish という Linux のコマンドライン ツールを使用して、仮想マシンの設定を QCOW2 ファイルに書き込みます。

前提条件

- KVM が構成されていること。「[「KVM のセットアップ」](#)」を参照してください。
- QCOW2 イメージを KVM ホストに展開する権限。

手順

- 1 `/var/lib/libvirt/images` ディレクトリに NSX Controller QCOW2 イメージをダウンロードします。
- 2 (Ubuntu のみ) 現在ログインしているユーザーを libvirtd ユーザーとして追加します。

```
adduser $USER libvirtd
```

- 3 QCOW2 イメージを保存したディレクトリに **guestinfo** というファイル（ファイル拡張子なし）を作成し、NSX Controller 仮想マシンのプロパティを入力します。

次はその例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="nsx_allowSSHRootLogin" oe:value="True"/>
    <Property oe:key="nsx_cli_audit_passwd_0"
oe:value="<password>"/>
    <Property oe:key="nsx_cli_passwd_0" oe:value="<password>"/>
    <Property oe:key="nsx_dns1_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_domain_0" oe:value="corp.local"/>
    <Property oe:key="nsx_gateway_0" oe:value="192.168.110.1"/>
    <Property oe:key="nsx_hostname" oe:value="nsx-Controller1"/>
    <Property oe:key="nsx_ip_0" oe:value="192.168.110.34"/>
    <Property oe:key="nsx_isSSHEnabled" oe:value="True"/>
    <Property oe:key="nsx_netmask_0" oe:value="255.255.255.0"/>
    <Property oe:key="nsx_ntp_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_passwd_0" oe:value="<password>"/>
  </PropertySection>
</Environment>
```

この例では、`nsx_isSSHEnabled` と `nsx_allowSSHRootLogin` がいずれも有効になっています。無効になっている場合、NSX Controller のコマンドラインへの SSH 接続やログインはできません。`nsx_isSSHEnabled` を有効にして、`nsx_allowSSHRootLogin` を有効にしなかった場合、NSX Controller に SSH で接続することはできませんが、`root` でログインすることはできません。

- 4 `guestfish` を使用して **guestinfo** ファイルを QCOW2 イメージに書き込みます。

複数の NSX Controller を作成する場合は、QCOW2 イメージのコピーをコントローラごとに作成します。**guestinfo** の情報を QCOW2 イメージに書き込んだ後、情報を上書きすることはできません。

```
sudo guestfish --rw -i -a nsx-controller1-build.qcow2 upload guestinfo /config/guestinfo
```

- 5 `virt-install` コマンドで QCOW2 イメージを展開します。

```
user@ubuntu1604:/var/lib/libvirt/images$ sudo virt-install --import --name nsx-  
controller1 --ram 16384 --vcpus 2 --network=bridge:br0,model=e1000 --disk  
path=/var/lib/libvirt/images/nsx-controller-<release_version_number>.qcow2,format=qcow2  
--nographics --noautoconsole
```

NSX Controller が起動したら、NSX Controller コンソールが表示されます。

- 6 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。

- 7 NSX-T コンポーネントのコンソールを開いて、ブート プロセスを追跡します。

- 8 NSX-T コンポーネントが起動した後、`admin` として CLI にログインし、`get interface eth0` コマンドを実行して、IP アドレスが想定どおりに適用されていることを確認します。

```
nsx-component> get interface eth0  
Interface: eth0  
Address: 192.168.110.25/24  
MAC address: 00:50:56:86:7b:1b  
MTU: 1500  
Default gateway: 192.168.110.1  
Broadcast address: 192.168.110.255  
...
```

- 9 NSX-T コンポーネントに必要な接続があることを確認します。

次のタスクを実行できることを確認します。

- 別のマシンから NSX-T コンポーネントに `ping` を実行します。
- NSX-T コンポーネントは、デフォルト ゲートウェイに `ping` を実行できます。
- NSX-T コンポーネントは、管理インターフェイスを使用して、NSX-T コンポーネントと同じネットワーク上のハイパーバイザー ホストに `ping` を実行できます。
- NSX-T コンポーネントは、DNS サーバと NTP サーバに `ping` を実行できます。
- SSH を有効にした場合は、SSH を使用して NSX-T コンポーネントに接続できることを確認します。

接続が確立されていない場合は、仮想アプライアンスのネットワーク アダプタが適切なネットワークまたは VLAN に配置されていることを確認します。

次のステップ

NSX Controller を管理プレーンに追加します。「[\[NSX Manager への NSX Controller の追加\]](#)」を参照してください。

NSX Manager への NSX Controller の追加

NSX Controller を NSX Manager に追加すると、NSX Manager と NSX Controller が相互に通信可能になります。

前提条件

- NSX Manager がインストールされていることを確認します。
- NSX Manager および NSX Controller アプライアンスにログインするための管理者権限を持っていることを確認します。

手順

- 1 NSX Manager への SSH セッションを開きます。
- 2 各 NSX Controller アプライアンスへの SSH セッションを開きます。
たとえば、NSX-Controller1、NSX-Controller2、NSX-Controller3 があるとします。
- 3 NSX Manager で **get certificate api thumbprint** コマンドを実行します。

```
NSX-Manager> get certificate api thumbprint
...
```

- 4 各 NSX Controller アプライアンスで [join management-plane] コマンドを実行します。

```
NSX-Controller1> join management-plane NSX-Manager-IP-address username admin
thumbprint <NSX-Manager-thumbprint>

Password for API user: <NSX-Manager-password>
Node successfully registered and controller restarted
```

展開された各 NSX Controller ノードに、このコマンドを実行します。

このとき、次の情報を指定します。

- NSX Manager の IP アドレスとオプションでポート番号
 - NSX Manager のユーザー名
 - NSX Manager の証明書サムプリント
 - NSX Manager のパスワード
- 5 NSX Controller で **get managers** コマンドを実行して結果を確認します。

```
NSX-Controller1> get managers
- 192.168.110.47    Connected
```

- 6 NSX Manager アプライアンスで `get management-cluster status` コマンドを実行して、NSX Controller が表示されることを確認します。

```
NSX-Manager> get management-cluster status
Number of nodes in management cluster: 1
- 192.168.110.47 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086) Online

Management cluster status: STABLE

Number of nodes in control cluster: 3
- 192.168.110.201 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086)
- 192.168.110.202 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086)
- 192.168.110.203 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086)
```

次のステップ

コントロール クラスタを初期化します。「[「コントロール クラスタの初期化によるコントロール クラスタ マスターの作成」](#)」を参照してください。

コントロール クラスタの初期化によるコントロール クラスタ マスターの作成

NSX-T 環境内に最初の NSX Controller をインストールしたら、コントロール クラスタを初期化できます。コントロール クラスタの初期化は、コントローラ ノードが 1 台のみの小規模な POC（事前検証）環境を構成する場合にも必要です。コントロール クラスタが初期化されないと、コントローラはハイパーバイザー ホストと通信できません。クラスタでは、1 つのコントローラのみを初期化する必要があります。

前提条件

- NSX Controller を 1 つ以上インストールします。
- NSX Controller を管理プレーンに追加します。
- NSX Controller アプライアンスにログインするための管理者権限を持っていることを確認します。
- 共有 Secret パスワードを割り当てます。共有 Secret パスワードは、ユーザー定義の共有 Secret パスワード（たとえば「secret123」）です。

手順

- 1 NSX Controller 用に SSH セッションを開きます。
- 2 `set control-cluster security-model shared-secret secret <secret>` コマンドを実行し、プロンプトが表示されたら共有 Secret を入力します。

3 initialize control-cluster コマンドを実行します。

このコマンドによって、このコントローラがコントロール クラスター マスターになります。

次はその例です。

```
NSX-Controller1> initialize control-cluster
Control cluster initialization successful.
```

4 get control-cluster status verbose コマンドを実行します。

is master と in majority が true、ステータスが active、Zookeeper Server IP が reachable, ok であることを確認します。

```
nsx-controller1> get control-cluster status verbose
NSX Controller Status:

uuid: 78d5b561-4f66-488d-9e53-089735eac1c1
is master: true
in majority: true
uuid                                address                status
78d5b561-4f66-488d-9e53-089735eac1c1 192.168.110.34         active

Cluster Management Server Status:

uuid                                rpc address            rpc port            global
id                                vpn address            status
557a911f-41fd-4977-9c58-f3ef55b3efe7 192.168.110.34         7777
1                                169.254.1.1            connected

Zookeeper Ensemble Status:

Zookeeper Server IP: 10.0.0.1, reachable, ok
Zookeeper version: 3.5.1-alpha--1, built on 03/08/2016 01:18 GMT
Latency min/avg/max: 0/0/1841
Received: 212095
Sent: 212125
Connections: 5
Outstanding: 0
Zxid: 0x10000017a
Mode: leader
Node count: 33
Connections: /10.0.0.1:51726[1]
(queued=0, recved=60324, sent=60324, sid=0x100000f14a10003, lop=PING, est=1459376913497, to=30000, lcxid=0x8, lzxid=0x10000017a, lresp=604617273, llat=0, minlat=0, avglat=0, maxlat=1088)
/10.0.0.1:35462[0] (queued=0, recved=1, sent=0)
/10.0.0.1:51724[1]
(queued=0, recved=45786, sent=45803, sid=0x100000f14a10001, lop=GETC, est=1459376911226, to=40000, lcxid=0x21e, lzxid=0x10000017a, lresp=604620658, llat=0, minlat=0, avglat=0, maxlat=1841)
/10.0.0.1:51725[1]
(queued=0, recved=60328, sent=60333, sid=0x100000f14a10002, lop=PING, est=1459376913455, to=30000, lcxid=0x8, lzxid=0x10000017a, lresp=604617273, llat=0, minlat=0, avglat=0, maxlat=1088)
```

```
000,lcxid=0xc,lzxid=0x10000017a,lresp=604618294,llat=0,minlat=0,avglat=0,maxlat=1356)
/10.0.0.1:51730[1]
(queued=0,recved=45315,sent=45324,sid=0x100000f14a10006,lop=PING,est=1459376914516,to=40
000,lcxid=0x49,lzxid=0x10000017a,lresp=604623243,llat=0,minlat=0,avglat=0,maxlat=1630)
```

次のステップ

コントロール クラスタにさらに NSX Controller を追加します。「[「クラスタ マスターを使用した NSX Controller の追加」](#)」を参照してください。

クラスタ マスターを使用した NSX Controller の追加

NSX Controller のマルチノード クラスタがあると、1 つ以上の NSX Controller が常に使用可能になります。

前提条件

- 3 台以上の NSX Controller アプライアンスをインストールします。
- NSX Controller アプライアンスにログインするための管理者権限を持っていることを確認します。
- NSX Controller のノードが管理プレーンに追加されていることを確認します。「[「NSX Manager への NSX Controller の追加」](#)」を参照してください。
- コントロール クラスタを初期化してコントロール クラスタ マスターを作成します。初期化する必要があるのは最初のコントローラだけです。
- **join control-cluster** コマンドでは、ドメイン名ではなく IP アドレスを使用する必要があります。
- vCenter Server を使用していて、NSX-T コントローラを同じクラスタに展開する場合は、DRS の非アフィニティ ルールを設定します。非アフィニティ ルールを設定すると、DRS で複数のノードが 1 台のホストに移行されることはありません。

手順

- 1 各 NSX Controller アプライアンス用に SSH セッションを開きます。

たとえば、NSX-Controller1、NSX-Controller2、NSX-Controller3 があるとします。この例では、NSX-Controller1 がコントロール クラスタを初期化済みで、コントロール クラスタ マスターになっています。

- 2 マスター以外の NSX Controller で、共有 Secret パスワードを指定して **set control-cluster security-model** コマンドを実行します。NSX-Controller2 と NSX-Controller3 で入力する共有 Secret パスワードは、NSX-Controller1 で入力した共有 Secret パスワードと同じである必要があります。

次はその例です。

```
NSX-Controller2> set control-cluster security-model shared-secret secret <NSX-Controller1's-shared-secret-password>
```

```
Security secret successfully set on the node.
```

```
NSX-Controller3> set control-cluster security-model shared-secret secret <NSX-Controller1's-shared-secret-password>
```

```
Security secret successfully set on the node.
```

- 3 マスター以外の NSX Controller で **get control-cluster certificate thumbprint** コマンドを実行します。

コマンド出力は、NSX Controller ごとに異なる一連の数値です。

次はその例です。

```
NSX-Controller2> get control-cluster certificate thumbprint
...
```

```
NSX-Controller3> get control-cluster certificate thumbprint
...
```

- 4 マスター NSX Controller で **[join control-cluster]** コマンドを実行します。

このとき、次の情報を指定します。

- IP アドレスと、オプションでマスター以外（この例では NSX-Controller2 と NSX-Controller3）の NSX Controller のポート番号
- マスター以外の NSX Controller の証明書のサムプリント

join コマンドは、複数のコントローラで並行して実行しないでください。追加処理が完了したことを確認してから、次のコントローラを追加します。

```
NSX-Controller1> join control-cluster <NSX-Controller2-IP> thumbprint <nsx-controller2's-thumbprint>
```

```
Node 192.168.210.48 has successfully joined the control cluster.
Please run 'activate control-cluster' command on the new node.
```

`get control-cluster status` コマンドを実行して、NSX-Controller2 がクラスタに追加されたことを確認します。

```
NSX-Controller1> join control-cluster <NSX-Controller3-IP> thumbprint <nsx-controller3's-thumbprint>
Node 192.168.210.49 has successfully joined the control cluster.
Please run 'activate control-cluster' command on the new node.
```

`get control-cluster status` コマンドを実行して、NSX-Controller3 がクラスタに追加されたことを確認します。

- 5 コントロールクラスタ マスターに追加された 2 台の NSX Controller ノードで `activate control-cluster` コマンドを実行します。

注: `activate` コマンドは、複数の NSX Controller で並行して実行しないでください。アクティベーション処理が完了したことを確認してから、次のコントローラのアクティベーションを行います。

次はその例です。

```
NSX-Controller2> activate control-cluster
Control cluster activation successful.
```

NSX-Controller2 で `get control-cluster status verbose` コマンドを実行し、Zookeeper Server IP が `reachable`, `ok` であることを確認します。

```
NSX-Controller3> activate control-cluster
Control cluster activation successful.
```

NSX-Controller3 で `get control-cluster status verbose` コマンドを実行し、Zookeeper Server IP が `reachable`, `ok` であることを確認します。

- 6 `get control-cluster status` コマンドを実行して結果を確認します。

```
NSX-Controller1> get control-cluster status
uuid: db4aa77a-4397-4d65-ad33-9fde79ac3c5c
is master: true
in majority: true


| uuid                                 | address        | status |
|--------------------------------------|----------------|--------|
| 0cfe232e-6c28-4fea-8aa4-b3518baef00d | 192.168.210.47 | active |
| bd257108-b94e-4e6d-8b19-7fa6c012961d | 192.168.210.48 | active |
| 538be554-1240-40e4-8e94-1497e963a2aa | 192.168.210.49 | active |


```

リストの最初の UUID は、コントローラ クラスタ全体を指しています。各 NSX Controller ノードにも UUID があります。

コントローラをクラスタに追加する際に、`set control-cluster security-model` または `join control-cluster` のいずれかのコマンドが失敗した場合は、クラスタの設定ファイルの整合性がとれていない可能性があります。

この問題を解決するには、次の手順を実行します。

- クラスタに追加しようとしている NSX Controller で **deactivate control-cluster** コマンドを実行します。
- マスター コントローラで、**get control-cluster status** または **get control-cluster status verbose** のいずれかのコマンドを実行すると、失敗したコントローラに関する情報が表示される場合は、**detach control-cluster <IP address of failed controller>** コマンドを実行します。

次のステップ

NSX Edge を展開します。「[章 6 「NSX Edge のインストール」](#)」を参照してください。

NSX Edge のインストール

NSX Edge は、ルーティングサービスと NSX-T 環境の外部のネットワークへの接続を提供します。ネットワーク アドレス変換 (NAT) や VPN などのステートフル サービスで Tier-0 ルーターまたは Tier-1 ルーターを展開する場合は、NSX Edge が必要です。

表 6-1. NSX Edge の展開、プラットフォームおよびインストール要件

要件	説明
サポートされる展開方法	<ul style="list-style-type: none"> ■ OVA/OVF ■ PXE を使用した ISO ■ PXE を使用しない ISO
サポート対象のプラットフォーム	NSX Edge は、ESXi またはベア メタルでのみサポートされます。NSX Edge は KVM ではサポートされていません。
PXE インストール	root ユーザーと admin ユーザーのパスワード文字列は、sha-512 アルゴリズムで暗号化する必要があります。
NSX-T アプライアンスのパスワード	<ul style="list-style-type: none"> ■ 8 文字以上 ■ 1 文字以上の小文字 ■ 1 文字以上の大文字 ■ 1 文字以上の数字 ■ 1 文字以上の特殊文字 ■ 5 文字以上の異なる文字 ■ 辞書に登録されている単語が使われていない ■ パリンドローム (回文) になっていない
ホスト名	NSX Edge をインストールするときに、アンダースコアなどの無効な文字を含まないホスト名を指定します。ホスト名に無効な文字が含まれていると、展開後にホスト名が localhost に設定されます。ホスト名の制限の詳細については、 https://tools.ietf.org/html/rfc952 および https://tools.ietf.org/html/rfc1123 を参照してください。
VMware Tools	ESXi で実行される NSX Edge 仮想マシンには、VMware Tools がインストールされています。VMware Tools を削除またはアップグレードしないでください。
システム	システム要件を満たしていることを確認します。「 システム要件 」を参照してください。

表 6-1. NSX Edge の展開、プラットフォームおよびインストール要件 (続き)

要件	説明
NSX のポート	<p>必要なポートが開いていることを確認します。「ポートとプロトコル」を参照してください。</p> <p>まだ作成していない場合は、宛先の仮想マシン ポート グループ ネットワークを作成します。管理仮想マシン ネットワークに NSX-T アプライアンスを配置することをお勧めします。</p>
IP アドレス	<p>複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。</p> <p>IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。</p> <p>IPv6 形式はサポートされていません。</p>
OVF テンプレート	<ul style="list-style-type: none"> ■ ESXi ホストに OVF テンプレートを展開するために必要な権限があることを確認します。 ■ ホスト名にアンダースコアが含まれていないことを確認します。そうでない場合は、ホスト名が <nsx-manager> に設定されます。 ■ OVF テンプレートを展開できる管理ツールが必要です (vCenter Server や vSphere Client など)。 <p>手動で設定できるようにするには、OVF 展開ツールで設定オプションがサポートされている必要があります。</p> <ul style="list-style-type: none"> ■ クライアント統合プラグインがインストールされている必要があります。
NTP サーバ	Edge クラスタ内のすべての NSX Edge サーバで同じ NTP サーバを設定する必要があります。

NSX Edge のインストール シナリオ

重要: vSphere Web Client またはコマンド ラインのいずれかを使用して OVA または OVF ファイルから NSX Edge をインストールすると、仮想マシンがパワーオン状態になるまで、ユーザー名、パスワード、IP アドレスなどの OVA/OVF プロパティ値が検証されません。

- **admin** または **audit** ユーザーのユーザー名を指定する場合には、一意の名前を使用する必要があります。同じ名前を指定すると、名前が無視され、デフォルトの名前 (**admin** または **audit**) が使用されます。
- **admin** ユーザーのパスワードが要件を満たしていない場合には、SSH またはコンソール経由で **admin** ユーザーとして NSX Edge にログインする必要があります。ログイン パスワードは **vmware** です。プロンプトが表示され、パスワードの変更が指示されます。
- **audit** ユーザーのパスワードが要件を満たしていない場合、ユーザー アカウントは無効になります。アカウントを有効にするには、**admin** ユーザーとして SSH またはコンソール経由で NSX Edge にログインし、**set user audit** コマンドを実行して **audit** ユーザーのパスワードを設定します (現在のパスワードは空の文字列です)。
- **root** ユーザーのパスワード要件を満たしていない場合には、**root** として SSH またはコンソール経由で NSX Edge にログインする必要があります。ログイン パスワードは **vmware** です。プロンプトが表示され、パスワードの変更が指示されます。

- この製品で操作を実行する場合、**root** ユーザーの認証情報を使用しないでください。このアクセスは、VMware サポート チームから要求された場合にのみ使用してください。**root** ユーザーの認証情報を使用してデーモンまたはアプリケーションをインストールすると、サポート契約が無効になります。

注: 要件を満たすパスワードが設定されるまで、コア サービスはアプライアンスで起動しません。

NSX Edge を OVA ファイルから展開した後は、仮想マシンをパワーオフして vCenter Server から OVA 設定を変更し、仮想マシンの IP アドレス設定を変更することはできません。

この章には、次のトピックが含まれています。

- [NSX Edge のネットワーク設定](#)
- [NSX Manager からの NSX Edge 仮想マシンの自動展開](#)
- [vSphere のグラフィカル ユーザー インターフェイス \(GUI\) を使用した ESXi への NSX Edge のインストール](#)
- [コマンドライン OVF ツールを使用した ESXi への NSX Edge のインストール](#)
- [PXE サーバで ISO ファイルを使用した NSX Edge のインストール](#)
- [ベア メタルへの NSX Edge のインストール](#)
- [ISO ファイルを使用した仮想アプライアンスとしての NSX Edge のインストール](#)
- [NSX Edge の管理プレーンへの追加](#)

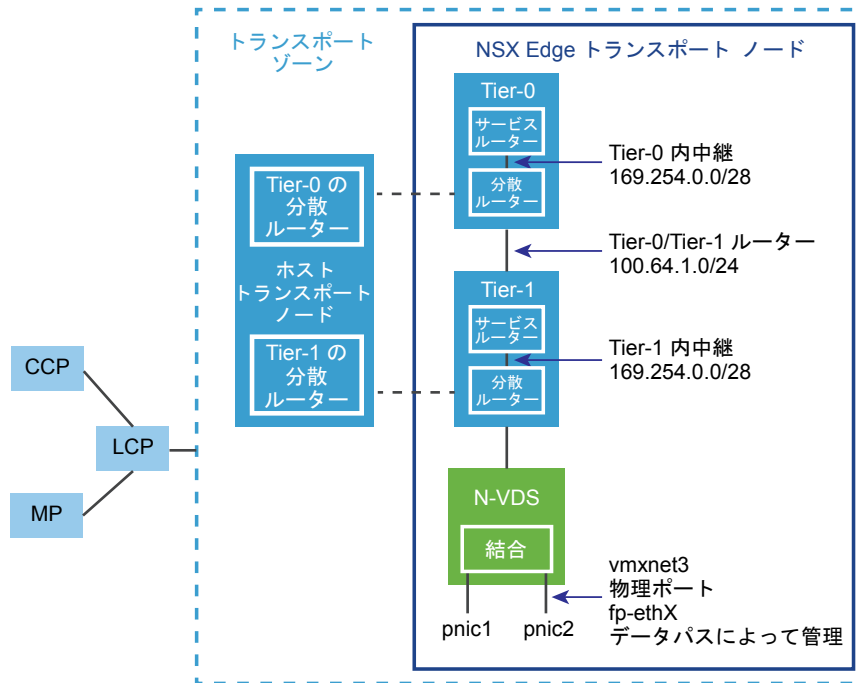
NSX Edge のネットワーク設定

NSX Edge は、ISO、OVA/OVF、または PXE を起動してインストールできます。いずれのインストール方法でも、NSX Edge をインストールする前にホスト ネットワークの準備ができていることを確認します。

トランスポート ゾーンにおける NSX Edge の概要図

NSX-T の概要図には、トランスポート ゾーンに 2 台のトランスポート ノードがあります。1 台のトランスポート ノードはホストです。もう 1 台は NSX Edge です。

図 6-1. NSX Edge の概要



展開直後の NSX Edge は、空のコンテナと考えることができます。論理ルーターを作成するまで、NSX Edge は何も行いません。NSX Edge は、Tier-0 と Tier-1 の論理ルーターの処理をバッキングします。各論理ルーターにはサービス ルーター (SR) と分散ルーター (DR) が含まれます。ルーターの分散とは、同じトランスポート ゾーンに属するすべてのトランスポート ノードにルーターを複製することです。この図では、ホストトランスポート ノードに、Tier-0 および Tier-1 と同じ分散ルーターが含まれています。サービス ルーターは、NAT などのサービスを実行するように論理ルーターを設定する場合に必要です。Tier-0 の論理ルーターにはすべてサービス ルーターがあります。Tier-1 のルーターには、設計上の検討事項に基づいて必要な場合にサービス ルーターを設定できます。

デフォルトでは、サービス ルーターと分散ルーター間のリンクは 169.254.0.0/28 サブネットを使用します。これらのルーター内の中継リンクは、Tier-0 または Tier-1 の論理ルーターの展開時に自動的に作成されます。環境内で 169.254.0.0/28 サブネットが使用中ではない限り、リンクを設定あるいは変更する必要はありません。Tier-1 の論理ルーターでは、この論理ルーターの作成時に NSX Edge クラスタを選択した場合にのみ SR があります。

Tier-0 から Tier-1 の接続に割り当てられるデフォルトのアドレス空間は 100.64.0.0/10 です。Tier-0 から Tier-1 の各ピア接続には、100.64.0.0/10 アドレス空間内で /31 サブネットが提供されます。このリンクは、Tier-1 ルーターを作成し、Tier-0 ルーターに接続するときに自動的に作成されます。環境内で 100.64.0.0/10 サブネットが使用中ではない限り、このリンクのインターフェイスを設定または変更する必要はありません。

NSX-T 環境には、それぞれ管理プレーン クラスタ (MP) と制御プレーン クラスタ (CCP) があります。管理プレーン クラスタと制御プレーン クラスタは、各トランスポート ゾーンのローカル制御プレーン (LCP) に設定をプッシュします。ホストまたは NSX Edge が管理プレーンに加わると、管理プレーン エージェント (MPA) がホストまたは NSX Edge と接続を確立し、ホストまたは NSX Edge が NSX-T のファブリック ノードになります。その後、ファブリック ノードがトランスポート ノードとして追加されると、ホストまたは NSX Edge との LCP 接続が確立されます。

この図は、高可用性を提供するために結合された2つの物理NIC (pNIC1とpNIC2) の例を示しています。物理NICはデータバスで管理されます。これらは、外部ネットワークへのVLAN アップリンクとして、または内部のNSX-Tで管理された仮想マシン ネットワークへのトンネル エンドポイントとして機能します。

ベスト プラクティスは、仮想マシンとして展開されている各 NSX Edge に 2 つ以上の物理リンクを割り当てることです。任意で、同じ pNIC のポート グループを、異なる VLAN ID を使用して重複させることができます。最初に見つかったネットワーク リンクが管理に使用されます。たとえば、NSX Edge 仮想マシンでは、vnic1 が最初に見つかったリンクだとします。ベア メタル インストールでは、eth0 または em0 が最初に見つかる場合があります。残りのリンクは、アップリンクやトンネルに使用されます。たとえば、1 つは、NSX-T によって管理されている仮想マシンのトンネルエンドポイントとして使用できます。もう 1 つは NSX Edge から外部 ToR へのアップリンクに使用できます。

管理者として CLI にログインし、コマンド **get interfaces** および **get physical-ports** を実行することによって、NSX Edge の物理リンク情報を表示できます。API では、**GET fabric/nodes/<edge-node-id>/network/interfaces** API 呼び出しを使用できます。物理リンクの詳細については、次のセクションを参照してください。

NSX Edge を仮想マシン アプライアンスとしてインストールするか、ベア メタルにインストールするかにかかわらず、ネットワーク設定には複数のオプションがあります。

トランスポート ゾーンと N-VDS

NSX Edge のネットワークを理解するには、トランスポート ゾーンと N-VDS についての知識が必要です。トランスポート ゾーンは NSX-T におけるレイヤー 2 ネットワークの到達範囲を制御します。N-VDS は、トランスポート ノードに作成されるソフトウェア スイッチです。N-VDS は、論理ルーターのアップリンクとダウンリンクを物理 NIC にバインドします。NSX Edge が属するトランスポート ゾーンごとに、NSX Edge に個別の N-VDS がインストールされます。

トランスポート ゾーンには次の 2 種類があります。

- トランスポート ノード間の内部 NSX-T トンネル用のオーバーレイ
- NSX-T 外部のアップリンク用 VLAN

NSX Edge はゼロ個の VLAN トランスポート ゾーンまたは多数のトランスポート ゾーンに属することができます。VLAN トランスポート ゾーンが 0 個の場合も、NSX Edge でアップリンクを使用できます。NSX Edge のアップリンクは、オーバーレイ トランスポート ゾーン用にインストールされている N-VDS を使用できるためです。各 NSX Edge に N-VDS を 1 つだけ設定する場合は、このようにできます。別の設計オプションとして、NSX Edge をアップリンクごとに 1 つずつ、複数の VLAN トランスポート ゾーンに加えることができます。

最も一般的な設計オプションは、3 つのトランスポート ゾーンです。1 つのオーバーレイと、冗長アップリンク用に 2 つの VLAN トランスポート ゾーンを設定します。

トランスポート ネットワークでオーバーレイ トラフィックに同じ VLAN ID を使用し、VLAN トラフィック (VLAN アップリンクなど) に別の VLAN ID を使用するには、2 つの異なる N-VDS (VLAN 用とオーバーレイ用) に ID を設定します。

トランスポート ゾーンの詳細については、「[トランスポート ゾーンについて](#)」を参照してください。

仮想アプライアンス/仮想マシンの NSX Edge ネットワーク

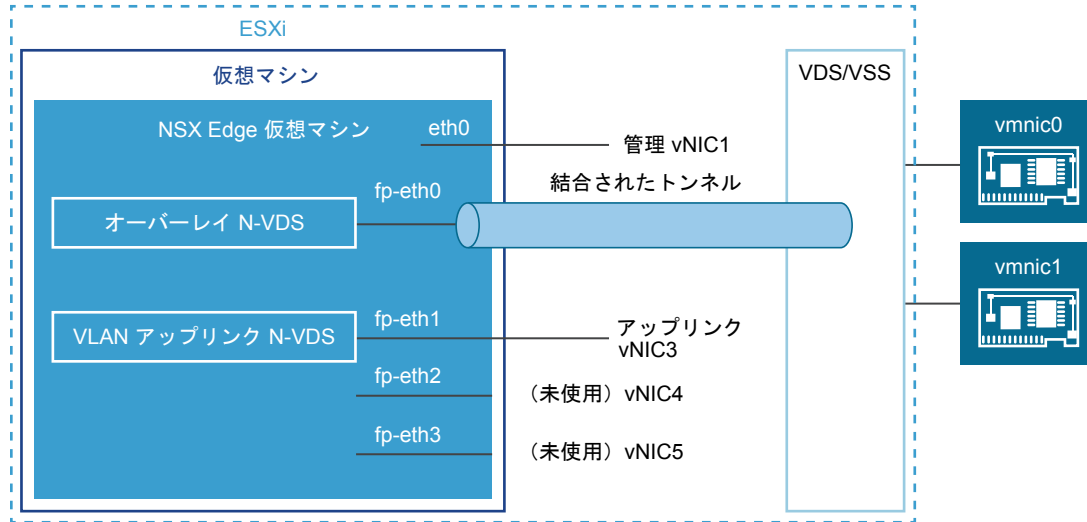
NSX Edge を仮想アプライアンスまたは仮想マシンとしてインストールすると、fp-ethX という内部インターフェイスが作成されます。ここで X は 0、1、2、3 です。これらのインターフェイスは、トップオブラック (ToR) スイッチへのアップリンク用と、NSX-T のオーバーレイ トンネル用に割り当てられます。

NSX Edge のトランスポート ノードを作成するときに、アップリンクとオーバーレイ トンネルに関連付ける fp-ethX インターフェイスを選択できます。fp-ethX インターフェイスの使用方法は任意に決められます。

vSphere Distributed Switch または vSphere Standard スイッチで、NSX Edge に 2 つ以上の vmnics を割り当てる必要があります。1 つは NSX Edge の管理用で、もう 1 つはアップリンクやトンネル用です。

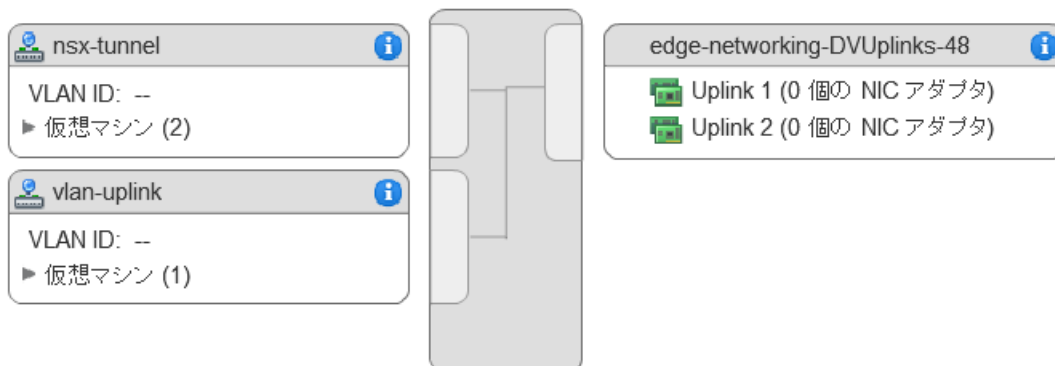
次の物理トポロジ例では、fp-eth0 を NSX-T のオーバーレイ トンネルに使用しています。fp-eth1 は VLAN アップリンクに使用しています。fp-eth2 と fp-eth3 は使用されていません。vNIC1 は、管理ネットワークに割り当てられます。

図 6-2. NSX Edge 仮想マシン ネットワークのリンク設定の例



この例に示す NSX Edge は 2 つのトランスポート ゾーンに属しています（オーバーレイ 1 つと VLAN 1 つ）。このため、トンネル用とアップリンク トラフィック用に 2 つの N-VDS があります。

このスクリーンショットは、仮想マシンのポート グループ、nsx-tunnel と vlan-uplink を示しています。



展開時には、仮想マシンのポート グループで設定されている名前と一致するネットワーク名を指定する必要があります。たとえば、NSX Edge の展開に ovftool を使用している場合に、この例の仮想マシン ポート グループと一致させるには、ネットワークの ovftool 設定を次のように行うことができます。

```
--net:"Network 0=Mgmt" --net:"Network 1=nsx-tunnel" --net:"Network 2=vlan-uplink"
```

この例では、仮想マシンのポート グループ名、Mgmt、nsx-tunnel、vlan-uplink を使用しています。仮想マシンのポート グループには任意の名前を使用できます。

NSX Edge でトンネルとアップリンク用に設定する仮想マシン ポート グループを、VMkernel ポートや、特定の IP アドレスに関連付ける必要はありません。これらは、レイヤー 2 でのみ使用されるためです。環境で DHCP を使用して管理インターフェイスにアドレスを提供する場合は、管理ネットワークに割り当てられている NIC が 1 つだけであることを確認します。

VLAN とトンネルのポート グループはトランク ポートとして設定されています。これは必須です。たとえば、標準の vSwitch では、トランク ポートを次のように設定します。[ホスト] - [設定] - [ネットワーク] - [ネットワークの追加] - [仮想マシン] - [VLAN ID すべて (4095)]。

アプライアンス ベースまたは仮想マシンの NSX Edge を使用する場合は、標準の vSwitch または vSphere Distributed Switch を使用できます。

準備が完了した NSX-T ホストに NSX Edge 仮想マシンをインストールして、トランスポート ノードとして設定することができます。展開には 2 つのタイプがあります。

- NSX Edge 仮想マシンを展開するには、VSS/VDS がホスト上の個別の pNIC を使用する VSS/VDS ポート グループを使用します。ホストトランスポート ノードは、ホストにインストールされた N-VDS に対して独立した pNIC を使用します。ホスト トランスポート ノードの N-VDS は、VSS または VDS と共存します。いずれも、独立した pNIC を使用します。ホスト TEP (Tunnel End Point) および NSX Edge TEP は同じサブネットに配置することも、異なるサブネットに配置することもできます。
- NSX Edge 仮想マシンを展開するには、ホスト トランスポート ノードの N-VDS 上の VLAN によってバックアップされている論理スイッチを使用します。ホスト TEP および NSX Edge TEP は異なるサブネットに配置する必要があります。

また、複数の NSX Edge アプライアンス/仮想マシンを 1 台のホストにインストールし、同じ管理、VLAN、トンネル エンドポイントのポート グループを、インストールされているすべての NSX Edge に使用できます。

基盤となる物理リンクが稼働し、仮想マシンのポート グループ設定が完了したら、NSX Edge をインストールできます。

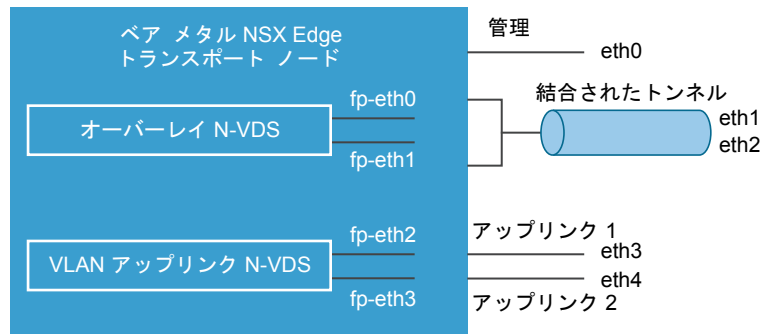
ベア メタルの NSX Edge ネットワーク

ベアメタルの NSX Edge には、fp-ethX という内部インターフェイスが含まれます。この X は、0、1、2、3、4 と続きます。作成される fp-ethX インターフェイス数は、ベア メタルの NSX Edge にある物理 NIC の数によって異なります。これらのインターフェイスの 4 つまでは、トップオブブラック (ToR) スイッチへのアップリンクや、NSX-T のオーバーレイ トンネルに割り当てることができます。

NSX Edge のトランスポート ノードを作成するときに、アップリンクとオーバーレイ トンネルに関連付ける fp-ethX インターフェイスを選択できます。

fp-ethX インターフェイスの使用方法は任意に決められます。次の物理トポロジ例では、fp-eth0 と fp-eth1 が結合され、NSX-T のオーバーレイ トンネルに使用されています。fp-eth2 と fp-eth3 は、ToR への冗長 VLAN アップリンクとして使用されています。

図 6-3. ペア メタルの NSX Edge ネットワークのリンク設定の一例



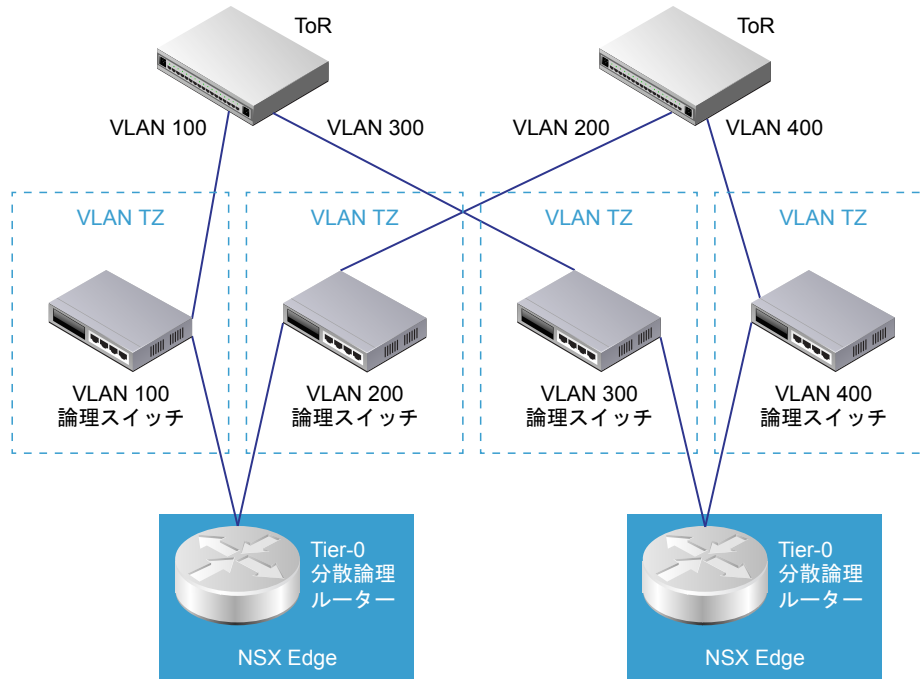
NSX Edge のアップリンクの冗長性

NSX Edge のアップリンクの冗長性によって、2 つの VLAN 等コスト マルチパス (ECMP) アップリンクを NSX Edge から外部 ToR のネットワーク接続に使用できます。

ECMP VLAN アップリンクが 2 つあるときは、高可用性と完全なメッシュ接続用に ToR スイッチも 2 つ用意する必要があります。VLAN 論理スイッチには、それぞれ対応する VLAN ID があります。

NSX Edge を VLAN のトランスポート ゾーンに追加すると、新しい N-VDS がインストールされます。たとえば、図に示すように 4 つの VLAN トランスポート ゾーンに NSX Edge ノードを追加すると、NSX Edge に 4 つの N-VDS がインストールされます。

図 6-4. NSX Edge から ToR への ECMP VLAN 設定の一例



注: vSphere Distributed Switch (vDS) が含まれ、N-VDS が含まれない ESXi ホスト上で展開されている Edge 仮想マシンの場合は、次の操作が必要です。

- 偽装転送を有効にして DHCP を機能させます。
- Edge 仮想マシンが不明なユニキャスト パケットを受信できるようにするため、無作為検出モードを有効にします。これは、MAC アドレスの学習がデフォルトで無効になっているためです。MAC アドレスの学習がデフォルトで有効になっている vDS 6.6 以降では、これは必要ありません。

NSX Manager からの NSX Edge 仮想マシンの自動展開

NSX Manager のユーザー インターフェイスで NSX Edge を構成し、vCenter Server に NSX Edge を自動的に展開できます。

前提条件

- [「NSX Edge のネットワーク設定」](#) で NSX Edge のネットワーク要件を参照してください。
- vCenter Server が NSX-T でコンピュート マネージャとして登録されている場合は、NSX Manager のユーザー インターフェイスを使用して、ホストを NSX Edge ノードとして設定し、自動的に vCenter Server に展開することができます。
- NSX Edge がインストールされている vCenter Server データストアで 120 GB 以上が使用可能であることを確認します。
- vCenter Server クラスタまたはホストが構成内で指定したネットワークとデータストアにアクセスできることを確認します。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック] - [ノード] - [Edge] - [Edge 仮想マシンの追加] の順に選択します。
- 3 NSX Edge の名前を入力します。
- 4 vCenter Server のホスト名または FQDN を入力します。
- 5 設定サイズとして、小、中、大のいずれかを選択します。
設定サイズによってシステム要件が異なります。
- 6 CLI とシステムの root パスワードを指定します。
root と CLI admin のパスワードに対する制限が自動展開にも適用されます。
- 7 ドロップダウン メニューからコンピュート マネージャを選択します。
コンピュート マネージャは、管理プレーンに登録されている vCenter Server です。
- 8 コンピュート マネージャに、ドロップダウン メニューからクラスタを選択するか、リソース プールを割り当てます。
- 9 NSX Edge 仮想マシンのファイルを格納するデータストアを選択します。
- 10 NSX Edge 仮想マシンを展開するクラスタを選択します。
ネットワーク管理機能を備えたクラスタに NSX Edge を追加することを推奨します。
- 11 ホストまたはリソース プールを選択します。一度に 1 台のホストのみを追加できます。
- 12 IP アドレスを選択して、管理ネットワークの IP アドレスと NSX Edge インターフェイスを配置するパスを入力します。IP アドレスは CIDR 形式で入力する必要があります。
管理ネットワークは、NSX Manager にアクセスできる必要があります。その IP アドレスを DHCP サーバから受信する必要があります。ネットワークは、NSX Edge の展開後に変更できます。
- 13 管理ネットワークの IP アドレスが NSX Manager ネットワークと同じレイヤー 2 に属していない場合には、デフォルト ゲートウェイを追加します。
NSX Manager と NSX Edge 管理ネットワーク間でレイヤー 3 接続が可能であることを確認します。

NSX Edge の展開が完了するまで 1 ～ 2 分かかります。展開状況は、ユーザー インターフェイスでリアルタイムに確認できます。

次のステップ

NSX Edge の展開に失敗した場合には、`/var/log/cm-inventory/cm-inventory.log` と `/var/log/proton/nsxapi.log` ファイルを参照して、問題を解決してください。

NSX Edge を NSX Edge クラスタに追加するか、トランスポート ノードとして構成する前に、新しく作成した NSX Edge ノードが「ノードの準備完了」と表示されていることを確認します。

vSphere のグラフィカル ユーザー インターフェイス (GUI) を使用した ESXi への NSX Edge のインストール

NSX Edge を対話形式でインストールする場合は、ユーザー インターフェイス ベースの仮想マシン管理ツールを使用できます。たとえば、vSphere Client を vCenter Server に接続して使用します。

NSX-T のこのリリースでは、IPv6 はサポートされていません。

前提条件

- [「NSX Edge のネットワーク設定」](#) で NSX Edge のネットワーク要件を参照してください。

手順

- 1 NSX Edge の OVA ファイルまたは OVF ファイルの場所を確認します。
ダウンロード URL をコピーするか、OVA ファイルをコンピュータにダウンロードします。
- 2 管理ツールで [OVF テンプレートの展開] ウィザードを起動し、.ova ファイルを指定します。
- 3 NSX Edge の名前を入力して、フォルダまたは vCenter Server データセンターを選択します。
ここに入力する名前がインベントリに表示されます。
選択したフォルダは、NSX Edge への権限の付与に使用します。
- 4 設定サイズとして、小、中、大のいずれかを選択します。
設定サイズによってシステム要件が異なります。『NSX-T リリース ノート』を参照してください。
- 5 NSX Edge の仮想アプライアンス ファイルを格納するデータストアを選択します。
- 6 vCenter Server にインストールする場合は、NSX Edge アプライアンスを展開するホストまたはクラスタを選択します。
通常は、ネットワーク管理機能を備えたクラスタに NSX Edge を配置します。
- 7 NSX Edge のインターフェイスを配置するネットワークを選択します。
ネットワークは、NSX Edge の展開後に変更できます。
- 8 NSX Edge のパスワードと IP アドレスを指定します。
- 9 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。
メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#) を参照してください。
- 10 NSX Edge のコンソールを開いて、ブート プロセスを追跡します。
コンソール ウィンドウが開かない場合は、ポップアップが許可されていることを確認してください。

- 11 NSX Edge の起動後、管理者権限で CLI にログインします。ユーザー名は **admin**、パスワードは **default** です。

注: NSX Edge の起動後、最初のログイン時に管理者認証情報を使用しなかった場合、データ プレーン サービスは NSX Edge で自動的に開始されません。

- 12 再起動後、管理者または root のいずれかの認証情報でログインできます。デフォルトの root パスワードは **vmware** です。
- 13 **get interface eth0** コマンドを実行して、IP アドレスが適切に適用されていることを確認します。

```
nsx-edge-1> [get interface eth0]

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

必要に応じて、**set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt** コマンドを実行して管理インターフェイスを更新します。オプションで、**start service ssh** コマンドで SSH サービスを起動できます。

- 14 NSX Edge アプライアンスで必要な接続が可能であることを確認します。

SSH を有効にした場合は、SSH を使用して NSX Edge に接続できることを確認します。

- NSX Edge に ping を実行できます。
- NSX Edge は、デフォルト ゲートウェイに ping を実行できます。
- NSX Edge は、NSX Edge と同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX Edge は、DNS サーバと NTP サーバに ping を実行できます。

- 15 接続問題のトラブルシューティングを行います。

注: 接続が確立されていない場合は、仮想マシン ネットワーク アダプタが適切なネットワークまたは VLAN に置かれていることを確認します。

デフォルトでは、NSX Edge データパスは、管理 NIC (IP アドレスとデフォルト ルートを持つ NIC) を除くすべての仮想マシン NIC を要求します。DHCP が管理機能に誤った NIC を割り当てた場合、問題を修正するタスクを実行します。

- a CLI にログインして **stop service dataplane** コマンドを入力します。
- b **set interface eth0 dhcp plane mgmt** コマンドを入力します。

- c eth0 を DHCP ネットワークに置き、IP アドレスが eth0 に割り当てられるまで待ちます。
- d **start service dataplane** コマンドを入力します。

VLAN アップリンクとトンネル オーバーレイに使用するデータパス fp-ethX ポートが、NSX Edge 上で [get interfaces] コマンドと [get physical-port] コマンドに示されます。

次のステップ

NSX Edge を管理プレーンに追加します。「[「NSX Edge の管理プレーンへの追加」](#)」を参照してください。

コマンドライン OVF ツールを使用した ESXi への NSX Edge のインストール

NSX Edge のインストールを自動的に行う場合は、コマンドライン ユーティリティの VMware OVF Tool を使用します。

NSX-T のこのリリースでは、IPv6 はサポートされていません。

前提条件

- システム要件を満たしていることを確認します。「[システム要件](#)」を参照してください。
- 必要なポートが開いていることを確認します。「[ポートとプロトコル](#)」を参照してください。
- まだ作成していない場合は、宛先の仮想マシン ポート グループ ネットワークを作成します。管理仮想マシン ネットワークに NSX-T アプライアンスを配置することをお勧めします。

複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。

- IPv4 IP アドレス スキームを使用します。NSX-T のこのリリースでは、IPv6 はサポートされていません。
- [「NSX Edge のネットワーク設定」](#) で NSX Edge のネットワーク要件を参照してください。
- ESXi ホストに OVF テンプレートを展開するために必要な権限があることを確認します。
- ホスト名にアンダースコアが含まれていないことを確認します。そうでない場合は、ホスト名が <localhost> に設定されます。
- OVF Tool バージョン 4.0 以降。

手順

- スタンドアロン ホストの場合、適切なパラメータを指定して **ovftool** コマンドを実行します。

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-edge-1
--deploymentOption=medium
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--net:"Network 0=Mgmt"
--net:"Network 1=nsx-tunnel"
```

```

--net:"Network 2=vlan-uplink"
--net:"Network 3=vlan-uplink"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.37
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=True
--prop:nsx_allowSSHRootLogin=True
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-edge
<path/url to nsx component ova>
vi://root:<password>@192.168.110.51

```

```

Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://root@192.168.110.24
Deploying to VI: vi://root@192.168.110.24
Transfer Completed
Powering on VM: nsx-edge-1
Task Completed
Completed successfully

```

- vCenter Server で管理されているホストの場合、適切なパラメータを指定して **ovftool** コマンドを実行します。

```

C:\Users\Administrator\Downloads>ovftool
--name=nsx-edge-1
--deploymentOption=medium
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--net:"Network 0=Mgmt"
--net:"Network 1=nsx-tunnel"
--net:"Network 2=vlan-uplink"
--net:"Network 3=vlan-uplink"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.37
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local

```

```
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=True
--prop:nsx_allowSSHRootLogin=True
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-edge
<path/url to nsx component ova>
vi://administrator@vsphere.local:<password>@192.168.110.24/?ip=192.168.210.53
```

```
Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://administrator@vsphere.local@192.168.110.24:443/
Deploying to VI: vi://administrator@vsphere.local@192.168.110.24:443/
Transfer Completed
Powering on VM: nsx-edge-1
Task Completed
Completed successfully
```

- (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。

- NSX Edge のコンソールを開いて、ブート プロセスを追跡します。
- NSX Edge の起動後、管理者権限で CLI にログインします。ユーザー名は **admin**、パスワードは **default** です。
- `get interface eth0` コマンドを実行して、IP アドレスが適切に適用されていることを確認します。

```
nsx-edge-1> [get interface eth0]

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

必要に応じて、`set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` コマンドを実行して管理インターフェイスを更新します。オプションで、`start service ssh` コマンドで SSH サービスを起動できます。

- NSX Edge アプライアンスで必要な接続が可能であることを確認します。
- SSH を有効にした場合は、SSH を使用して NSX Edge に接続できることを確認します。
- NSX Edge に ping を実行できます。

- NSX Edge は、デフォルト ゲートウェイに ping を実行できます。
- NSX Edge は、NSX Edge と同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX Edge は、DNS サーバと NTP サーバに ping を実行できます。
- 接続問題のトラブルシューティングを行います。

注: 接続が確立されていない場合は、仮想マシン ネットワーク アダプタが適切なネットワークまたは VLAN に置かれていることを確認します。

デフォルトでは、NSX Edge データパスは、管理 NIC (IP アドレスとデフォルト ルートを持つ NIC) を除くすべての仮想マシン NIC を要求します。DHCP が管理機能に誤った NIC を割り当てた場合、問題を修正するタスクを実行します。

- a CLI にログインして **stop service dataplane** コマンドを入力します。
- b **set interface eth0 dhcp plane mgmt** コマンドを入力します。
- c eth0 を DHCP ネットワークに置き、IP アドレスが eth0 に割り当てられるまで待ちます。
- d **start service dataplane** コマンドを入力します。

VLAN アップリンクとトンネル オーバーレイに使用するデータパス fp-ethX ポートが、NSX Edge 上で [get interfaces] コマンドと [get physical-port] コマンドに示されます。

次のステップ

NSX Edge を管理プレーンに追加します。「[\[NSX Edge の管理プレーンへの追加\]](#)」を参照してください。

PXE サーバで ISO ファイルを使用した NSX Edge のインストール

NSX Edge デバイスは、PXE を使用して、ベア メタル上または仮想マシンとして自動的にインストールできます。PXE ブートのインストールは、NSX Manager と NSX Controller ではサポートされていません。このとき、IP アドレス、ゲートウェイ、ネットワーク マスク、NTP、DNS などのネットワーク設定が自動的に行われます。

この手順で、Ubuntu で PXE サーバのセットアップを実行します。PXE は DHCP、HTTP、TFTP の複数のコンポーネントから構成されます。

DHCP は、NSX Edge などの NSX-T コンポーネントに IP アドレス設定を動的に配信します。PXE 環境の DHCP サーバでは、NSX Edge が IP アドレスを自動的に要求し、受け取ることができます。

TFTP はファイル転送プロトコルです。TFTP サーバは、ネットワーク上で常に PXE クライアントを待機しています。PXE サービスを要求するネットワーク PXE クライアントが検出されると、NSX-T コンポーネントの ISO ファイルと、preseed ファイルに含まれるインストール設定が提供されます。

PXE サーバの準備ができた後、preseed の設定ファイルを使用して NSX Edge をインストールする手順は次のとおりです。

前提条件

- 環境で PXE サーバが使用できる必要があります。PXE サーバは任意の Linux ディストリビューションに設定できます。PXE サーバには 2 つのインターフェイスが必要です。1 つは外部通信用で、もう 1 つは DHCP の IP アドレス サービスと TFTP サービス用です。

複数の管理ネットワークが存在する場合は、NSX-T アプライアンスから他のネットワークへのスタティック ルートを追加できます。

- [「NSX Edge のネットワーク設定」](#) で NSX Edge のネットワーク要件を参照してください。

手順

- 1 (オプション) kickstart ファイルを使用して、Ubuntu サーバで新しい TFTP または DHCP サービスをセットアップします。

kickstart ファイルはテキスト ファイルで、最初の起動後にアプライアンスで実行する CLI コマンドが含まれます。

参照する PXE サーバに基づいて、kickstart ファイルに名前を付けます。次はその例です。

```
nsxcli.install
```

という名前で、Web サーバの `/var/www/html/nsx-edge/nsxcli.install` などにコピーする必要があります。

kickstart ファイルに、CLI コマンドを追加できます。

次はその例です。

管理インターフェイスの IP アドレスを設定するには、次のコマンドを使用します。

```
stop dataplane
set interface eth0 <ip-cidr-format> plane mgmt
start dataplane
```

admin ユーザーのパスワードを変更するには、次のコマンドを使用します。

```
set user admin password <new_password> old-password <old-password>
```

preseed.cfg ファイルでパスワードを指定する場合は、kickstart ファイルでも同じパスワードを使用します。それ以外の場合は、デフォルトのパスワードである「default」を使用します。

NSX Edge を管理プレーンに追加するには、次のコマンドを使用します。

```
join management-plane <mgr-ip> thumbprint <mgr-thumbprint> username <mgr-username>
password <mgr password>
```

- 2 2 つのインターフェイスを作成します。1 つは管理用で、もう 1 つは DHCP サービスと TFTP サービス用です。DHCP/TFTP インターフェイスが、NSX Edge を配置する予定のサブネットにあることを確認します。

たとえば、NSX Edge の管理インターフェイスを 192.168.210.0/24 サブネットに配置する場合は、eth1 を同じサブネットに配置します。

```
# The loopback network interface
auto lo
iface lo inet loopback

# PXE server's management interface
auto eth0
iface eth0 inet static
    address 192.168.110.81
    gateway 192.168.110.1
    netmask 255.255.255.0
    dns-nameservers 192.168.110.10

# PXE server's DHCP/TFTP interface
auto eth1
iface eth1 inet static
    address 192.168.210.82
    gateway 192.168.210.1
    netmask 255.255.255.0
    dns-nameservers 192.168.110.10
```

- 3 DHCP サーバソフトウェアをインストールします。

```
sudo apt-get install isc-dhcp-server -y
```

- 4 `/etc/default/isc-dhcp-server` ファイルを編集し、DHCP サービスを提供するインターフェイスを追加します。

```
INTERFACES="eth1"
```

- 5 (オプション) この DHCP サーバをローカル ネットワークの正式な DHCP サーバにする場合は、`/etc/dhcp/dhcpd.conf` ファイルで `[authoritative;]` 行をコメント解除します。

```
...
authoritative;
...
```

- 6 `/etc/dhcp/dhcpd.conf` で、PXE ネットワークの DHCP 設定を定義します。

次はその例です。

```
subnet 192.168.210.0 netmask 255.255.255.0 {
    range 192.168.210.90 192.168.210.95;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 192.168.110.10;
    option routers 192.168.210.1;
```

```
option broadcast-address 192.168.210.255;
default-lease-time 600;
max-lease-time 7200;
}
```

- 7 DHCP サービスを開始します。

```
sudo service isc-dhcp-server start
```

- 8 DHCP サービスが動作することを確認してください。

```
service --status-all | grep dhcp
```

- 9 Apache、TFTP、PXE ブートに必要なその他のコンポーネントをインストールします。

```
sudo apt-get install apache2 tftpd-hpa inetutils-inetd
```

- 10 TFTP と Apache が実行されていることを確認します。

```
service --status-all | grep tftpd-hpa
service --status-all | grep apache2
```

- 11 次の行を `/etc/default/tftpd-hpa` ファイルに追加します。

```
RUN_DAEMON="yes"
OPTIONS="-l -s /var/lib/tftpboot"
```

- 12 次の行を `/etc/inetd.conf` ファイルに追加します。

```
tftp dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd -
s /var/lib/tftpboot
```

- 13 TFTP サービスを再起動します。

```
sudo /etc/init.d/tftpd-hpa restart
```

- 14 NSX Edge インストーラ ISO ファイルを一時フォルダにコピーするかダウンロードします。

- 15 ISO ファイルをマウントし、インストール コンポーネントを TFTP サーバと Apache サーバにコピーします。

```
sudo mount -o loop ~/nsx-edge.<build>.iso /mnt
cd /mnt
sudo cp -fr install/netboot/* /var/lib/tftpboot/
sudo mkdir /var/www/html/nsx-edge
sudo cp -fr /mnt/* /var/www/html/nsx-edge/
```

- 16 (オプション) `/var/www/html/nsx-edge/preseed.cfg` ファイルを編集して、暗号化されているパスワードを変更します。

`mkpasswd` などの Linux ツールを使用してパスワード ハッシュを作成できます。

```
sudo apt-get install whois
sudo mkpasswd -m sha-512
```

```
Password:
$6$SUFQs[...]FcoHLijOuFD
```

- a root パスワードを変更するには、`/var/www/html/nsx-edge/preseed.cfg` で次の行を検索します。

```
d-i passwd/root-password-crypted password $6$tgmLNLMP$9BuAHhN...
```

- b ハッシュ文字列を置換します。

`$`、`'`、`"`、`\`などの特殊文字をエスケープする必要はありません。

- c `usermod` コマンドを `preseed.cfg` に追加して、root または admin、あるいはその両方のパスワードを設定します。

たとえば、`echo 'VMware NSX Edge'` 行を検索して、次のコマンドを追加します。

```
usermod --password '\$6\$VS3exId0aKmzW\$U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/'
root; \
usermod --password '\$6\$VS3exId0aKmzW\$U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/'
admin; \
```

このハッシュ文字列は一例です。特殊文字はすべてエスケープする必要があります。最初の `usermod` コマンドの root パスワードが、`d-i passwd/root-password-crypted password 6tgm...` で設定したパスワードに置き換わります。

`usermod` コマンドを使用してパスワードを設定した場合、ユーザーは初めてログインするときにパスワードの変更を求められません。それ以外の場合、ユーザーは初回のログイン時にパスワードを変更する必要があります。

- 17 次の行を `/var/lib/tftpboot/pxelinux.cfg/default` ファイルに追加します。

192.168.210.82 は、実際の TFTP サーバの IP アドレスに置き換えます。

```
label nsxedge
    kernel ubuntu-installer/amd64/linux
    ipappend 2
    append netcfg/dhcp_timeout=60 auto=true priority=critical vga=normal partman-
lvm/device_remove_lvm=true netcfg/choose_interface=auto debian-
installer/allow_unauthenticated=true preseed/url=http://192.168.210.82/nsx-
edge/preseed.cfg mirror/country=manual mirror/http/hostname=192.168.210.82 nsx-
kickstart/url=http://192.168.210.82/nsx-edge/nsxcli.install mirror/http/directory=/nsx-
edge initrd=ubuntu-installer/amd64/initrd.gz mirror/suite=xenial --
```

- 18 次の行を `/etc/dhcp/dhcpd.conf` ファイルに追加します。

192.168.210.82 は、実際の DHCP サーバの IP アドレスに置き換えます。

```
allow booting;
allow bootp;

next-server 192.168.210.82; #Replace this IP address
filename "pxelinux.0";
```

- 19 DHCP サービスを再起動します。

```
sudo service isc-dhcp-server restart
```

注: 「stop: Unknown instance: start: Job failed to start」などのエラーが返された場合、`sudo /etc/init.d/isc-dhcp-server stop` を実行してから `sudo /etc/init.d/isc-dhcp-server start` を実行します。`sudo /etc/init.d/isc-dhcp-server start` コマンドは、エラーの原因に関する情報を返します。

- 20 ベア メタルのインストール手順または ISO のインストール手順に従ってインストールを完了します。

- [「ベア メタルへの NSX Edge のインストール」](#)
- [「ISO ファイルを使用した仮想アプライアンスとしての NSX Edge のインストール」](#)

- 21 NSX 仮想マシンまたは NSX ベア メタル ホストをパワーオンします。

- 22 ブート メニューで [nsxedge] を選択します。

ネットワークが設定され、パーティションが作成されて、NSX Edge コンポーネントがインストールされます。

NSX Edge のログイン プロンプトが表示されたら、admin または root でログインできます。

デフォルトでは、root のログインパスワードは [vmware] で、admin のログインパスワードは [default] です。

- 23 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#) を参照してください。

- 24 NSX Edge のコンソールを開いて、ブート プロセスを追跡します。

コンソール ウィンドウが開かない場合は、ポップアップが許可されていることを確認してください。

- 25 NSX Edge の起動後、管理者権限で CLI にログインします。ユーザー名は **admin**、パスワードは **default** です。

注: NSX Edge の起動後、最初のログイン時に管理者認証情報を使用しなかった場合、データ プレーン サービスは NSX Edge で自動的に開始されません。

- 26 再起動後、管理者または root のいずれかの認証情報でログインできます。デフォルトの root パスワードは **vmware** です。

27 `get interface eth0` コマンドを実行して、IP アドレスが適切に適用されていることを確認します。

```
nsx-edge-1> [get interface eth0]

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

必要に応じて、`set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` コマンドを実行して管理インターフェイスを更新します。オプションで、`start service ssh` コマンドで SSH サービスを起動できます。

28 NSX Edge アプライアンスで必要な接続が可能であることを確認します。

SSH を有効にした場合は、SSH を使用して NSX Edge に接続できることを確認します。

- NSX Edge に ping を実行できます。
- NSX Edge は、デフォルト ゲートウェイに ping を実行できます。
- NSX Edge は、NSX Edge と同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX Edge は、DNS サーバと NTP サーバに ping を実行できます。

29 接続問題のトラブルシューティングを行います。

注: 接続が確立されていない場合は、仮想マシン ネットワーク アダプタが適切なネットワークまたは VLAN に置かれていることを確認します。

デフォルトでは、NSX Edge データパスは、管理 NIC (IP アドレスとデフォルト ルートを持つ NIC) を除くすべての仮想マシン NIC を要求します。DHCP が管理機能に誤った NIC を割り当てた場合、問題を修正するタスクを実行します。

- a CLI にログインして **stop service dataplane** コマンドを入力します。
- b **set interface eth0 dhcp plane mgmt** コマンドを入力します。
- c eth0 を DHCP ネットワークに置き、IP アドレスが eth0 に割り当てられるまで待ちます。
- d **start service dataplane** コマンドを入力します。

VLAN アップリンクとトンネル オーバーレイに使用するデータパス fp-ethX ポートが、NSX Edge 上で [get interfaces] コマンドと [get physical-port] コマンドに示されます。

次のステップ

NSX Edge を管理プレーンに追加します。[「NSX Edge の管理プレーンへの追加」](#) を参照してください。

ベア メタルへの NSX Edge のインストール

NSX Edge デバイスを手動でベア メタルにインストールするには、ISO ファイルを使用します。このファイルには、IP アドレス、ゲートウェイ、ネットワーク マスク、NTP、DNS などのネットワーク設定が含まれます。

前提条件

- システム BIOS モードがレガシー BIOS に設定されていることを確認します。
- [「NSX Edge のネットワーク設定」](#) で NSX Edge のネットワーク要件を参照してください。

手順

- 1 起動可能なディスクを作成し、NSX Edge の ISO ファイルを置きます。
- 2 ディスクから物理マシンを起動します。
- 3 [自動インストール] を選択します。

Enter キーを押した後、開始するまでに 10 秒程かかる可能性があります。

パワーオン中に、インストーラから DHCP を介したネットワーク設定を求められます。環境内で DHCP を使用できない場合は、インストーラに IP アドレスの設定を求めるプロンプトが表示されます。

デフォルトでは、root のログインパスワードは [vmware] で、admin のログインパスワードは [default] です。

- 4 NSX Edge のコンソールを開いて、ブート プロセスを追跡します。

コンソール ウィンドウが開かない場合は、ポップアップが許可されていることを確認してください。

- 5 NSX Edge の起動後、管理者権限で CLI にログインします。ユーザー名は **admin**、パスワードは **default** です。

注: NSX Edge の起動後、最初のログイン時に管理者認証情報を使用しなかった場合、データ プレーン サービスは NSX Edge で自動的に開始されません。

- 6 再起動後、管理者または root のいずれかの認証情報でログインできます。デフォルトの root パスワードは **vmware** です。

- 7 `get interface eth0` コマンドを実行して、IP アドレスが適切に適用されていることを確認します。

```
nsx-edge-1> [get interface eth0]
```

```
Interface: eth0
```

```
Address: 192.168.110.37/24
```

```
MAC address: 00:50:56:86:62:4d
```

```
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

必要に応じて、**set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt** コマンドを実行して管理インターフェイスを更新します。オプションで、**start service ssh** コマンドで SSH サービスを起動できます。

8 NSX Edge アプライアンスで必要な接続が可能であることを確認します。

SSH を有効にした場合は、SSH を使用して NSX Edge に接続できることを確認します。

- NSX Edge に ping を実行できます。
- NSX Edge は、デフォルト ゲートウェイに ping を実行できます。
- NSX Edge は、NSX Edge と同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX Edge は、DNS サーバと NTP サーバに ping を実行できます。

9 接続問題のトラブルシューティングを行います。

注: 接続が確立されていない場合は、仮想マシン ネットワーク アダプタが適切なネットワークまたは VLAN に置かれていることを確認します。

デフォルトでは、NSX Edge データパスは、管理 NIC (IP アドレスとデフォルト ルートを持つ NIC) を除くすべての仮想マシン NIC を要求します。DHCP が管理機能に誤った NIC を割り当てた場合、問題を修正するタスクを実行します。

- a CLI にログインして **stop service dataplane** コマンドを入力します。
- b **set interface eth0 dhcp plane mgmt** コマンドを入力します。
- c eth0 を DHCP ネットワークに置き、IP アドレスが eth0 に割り当てられるまで待ちます。
- d **start service dataplane** コマンドを入力します。

VLAN アップリンクとトンネル オーバーレイに使用するデータパス fp-ethX ポートが、NSX Edge 上で [get interfaces] コマンドと [get physical-port] コマンドに示されます。

次のステップ

NSX Edge を管理プレーンに追加します。「[「NSX Edge の管理プレーンへの追加」](#)」を参照してください。

ISO ファイルを使用した仮想アプライアンスとしての NSX Edge のインストール

NSX Edge 仮想マシンは、ISO ファイルを使用して手動でインストールできます。

重要: NSX-T コンポーネント仮想マシンのインストールには VMware Tools が含まれます。NSX-T アプライアンスで VMware Tools を削除またはアップグレードすることはできません。

前提条件

- 「NSX Edge のネットワーク設定」で NSX Edge のネットワーク要件を参照してください。

手順

- 1 スタンドアロン ホストまたは vSphere Web Client で仮想マシンを作成し、次のリソースを割り当てます。

- ゲスト OS : その他 (64 ビット)
- VMXNET3 NIC×3。NSX Edge では e1000 NIC ドライバはサポートされません。
- NSX-T 環境に必要なシステム リソース。

- 2 NSX Edge の ISO ファイルを仮想マシンにバインドします。

CD/DVD ドライブのデバイスの状態が [パワーオン時に接続] に設定されていることを確認します。

edge-from-iso - 設定の編集	
仮想ハードウェア	仮想マシン オプション Storage DRS ルール vApp オプション
CPU	1
メモリ	2048 MB
ハード ディスク 1	16 GB
SCSI コントローラ 0	VMware 準仮想化
ネットワーク アダプタ 1	nsx-tunnel (edge-networking) <input checked="" type="checkbox"/> 接続...
*CD/DVD ドライブ 1	データストア ISO ファイル
ステータス	<input checked="" type="checkbox"/> パワーオン時に接続
CD/DVD メディア	[datastore (2)]/nsx-edge-2.3 参照...
デバイス モード	パススルー CD-ROM
仮想デバイス ノード	SATA コントローラ 0 SATA(0:0)
フロッピー ドライブ 1	クライアント デバイス <input type="checkbox"/> 接続...
ビデオ カード	カスタム設定の指定
SATA コントローラ 0	
VMCI デバイス	
その他のデバイス	

- 3 ISO の起動時に、仮想マシン コンソールを開いて [自動インストール] を選択します。

Enter キーを押した後、開始するまでに 10 秒程かかる可能性があります。

パワーオン中に、仮想マシンが DHCP を介したネットワーク設定を要求します。環境内で DHCP を使用できない場合は、インストーラに IP アドレスの設定を求めるプロンプトが表示されます。

デフォルトでは、root のログイン パスワードは [vmware] で、admin のログインパスワードは [default] です。

初回ログイン時にパスワードの変更を求められます。このパスワードの変更には、次に示すような厳密な複雑性ルールが適用されます。

- 8 文字以上
- 1 文字以上の小文字
- 1 文字以上の大文字
- 1 文字以上の数字
- 1 文字以上の特殊文字
- 5 文字以上の異なる文字
- 辞書に登録されている単語が使われていない
- パリンドローム（回文）になっていない

重要: 要件を満たすパスワードが設定されるまで、コア サービスはアプライアンスで起動しません。

- 4 (オプション) 最適なパフォーマンスを実現するように、NSX-T コンポーネント用のメモリを予約します。

メモリ予約は、ホストが仮想マシン用に予約する物理メモリ容量の確実な下限であり、メモリがオーバーコミットされる場合でも、この容量が確保されます。NSX-T コンポーネントが効率的に動作するのに十分なメモリが確保されるように、予約のレベルを設定します。[「システム要件」](#)を参照してください。

- 5 NSX Edge のコンソールを開いて、ブート プロセスを追跡します。

コンソール ウィンドウが開かない場合は、ポップアップが許可されていることを確認してください。

- 6 NSX Edge の起動後、管理者権限で CLI にログインします。ユーザー名は **admin**、パスワードは **default** です。

注: NSX Edge の起動後、最初のログイン時に管理者認証情報を使用しなかった場合、データ プレーン サービスは NSX Edge で自動的に開始されません。

- 7 再起動後、管理者または root のいずれかの認証情報でログインできます。デフォルトの root パスワードは **vmware** です。

- 8 `get interface eth0` コマンドを実行して、IP アドレスが適切に適用されていることを確認します。

```
nsx-edge-1> [get interface eth0]

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

必要に応じて、`set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` コマンドを実行して管理インターフェイスを更新します。オプションで、`start service ssh` コマンドで SSH サービスを起動できます。

9 NSX Edge アプライアンスで必要な接続が可能であることを確認します。

SSH を有効にした場合は、SSH を使用して NSX Edge に接続できることを確認します。

- NSX Edge に ping を実行できます。
- NSX Edge は、デフォルト ゲートウェイに ping を実行できます。
- NSX Edge は、NSX Edge と同じネットワーク上のハイパーバイザー ホストに ping を実行できます。
- NSX Edge は、DNS サーバと NTP サーバに ping を実行できます。

10 接続問題のトラブルシューティングを行います。

注: 接続が確立されていない場合は、仮想マシン ネットワーク アダプタが適切なネットワークまたは VLAN に置かれていることを確認します。

デフォルトでは、NSX Edge データパスは、管理 NIC (IP アドレスとデフォルト ルートを持つ NIC) を除くすべての仮想マシン NIC を要求します。DHCP が管理機能に誤った NIC を割り当てた場合、問題を修正するタスクを実行します。

- a CLI にログインして **stop service dataplane** コマンドを入力します。
- b **set interface eth0 dhcp plane mgmt** コマンドを入力します。
- c eth0 を DHCP ネットワークに置き、IP アドレスが eth0 に割り当てられるまで待ちます。
- d **start service dataplane** コマンドを入力します。

VLAN アップリンクとトンネル オーバーレイに使用するデータパス fp-ethX ポートが、NSX Edge 上で [get interfaces] コマンドと [get physical-port] コマンドに示されます。

次のステップ

NSX Edge を管理プレーンに追加します。「[\[NSX Edge の管理プレーンへの追加\]](#)」を参照してください。

NSX Edge の管理プレーンへの追加

NSX Edge を管理プレーンに追加すると、NSX Manager と NSX Edge が相互に通信できるようになります。

前提条件

NSX Edge および NSX Manager アプライアンスにログインするための管理者権限を持っていることを確認します。

手順

- 1 NSX Manager アプライアンスへの SSH セッションを開きます。
- 2 NSX Edge への SSH セッションを開きます。

3 NSX Manager アプライアンスで `get certificate api thumbprint` コマンドを実行します。

コマンド出力は、この NSX Manager に固有の一連の英数字です。

次はその例です。

```
NSX-Manager1> get certificate api thumbprint
...
```

4 NSX Edge で `[join management-plane]` コマンドを実行します。

このとき、次の情報を指定します。

- NSX Manager のホスト名または IP アドレスとオプションでポート番号
- NSX Manager のユーザー名
- NSX Manager の証明書サムプリント
- NSX Manager のパスワード

```
NSX-Edge1> join management-plane NSX-Manager1 username admin thumbprint <NSX-Manager1's-thumbprint>
Password for API user: <NSX-Manager1's-password>
Node successfully registered and Edge restarted
```

このコマンドを各 NSX Edge ノードで繰り返します。

NSX Edge で `get managers` コマンドを実行して結果を確認します。

```
nsx-edge-1> get managers
- 192.168.110.47    Connected
```

NSX Manager のユーザー インターフェイスで、NSX Edge が [ファブリック] > [Edge] ページに表示されます。MPA の接続状態は「稼動中」になります。MPA の接続状態が「稼動中」ではない場合は、ブラウザ画面を更新してみます。

次のステップ

NSX Edge をトランスポート ノードとして追加します。「[「NSX Edge トランスポート ノードの作成」](#)」を参照してください。

ホストの準備

NSX-T と連携する準備ができたハイパーバイザー ホストをファブリック ノードと呼びます。ファブリック ノードとなったホストは、NSX-T モジュールがインストールされ、NSX-T 管理プレーンに登録されています。

この章には、次のトピックが含まれています。

- [KVM ホストへのサードパーティ製パッケージのインストール](#)
- [RHEL KVM ホストの Open vSwitch のバージョンを確認する](#)
- [NSX-T ファブリックへのハイパーバイザー ホストの追加](#)
- [NSX-T カーネル モジュールの手動インストール](#)
- [ハイパーバイザー ホストの管理プレーンへの追加](#)

KVM ホストへのサードパーティ製パッケージのインストール

KVM ホストをファブリック ノードにする準備を整えるには、いくつかのサードパーティ製パッケージをインストールする必要があります。

前提条件

(Red Hat) サードパーティのパッケージをインストールする前に、仮想化パッケージをインストールします。Red Hat ホストで、次のコマンドを実行します。

```
yum groupinstall "Virtualization Hypervisor"  
yum groupinstall "Virtualization Client"  
yum groupinstall "Virtualization Platform"  
yum groupinstall "Virtualization Tools"
```

パッケージをインストールできない場合は、コマンド `yum install glibc.i686 nspr` を使用して、Red Hat を新たに手動でインストールできます。

(Ubuntu) サードパーティのパッケージをインストールする前に、仮想化パッケージをインストールします。Ubuntu ホストで、次のコマンドを実行します。

```
apt-get install qemu-kvm
apt-get install libvirt-bin
apt-get install virtinst
apt-get install virt-manager
apt-get install virt-viewer
apt-get install ubuntu-vm-builder
apt-get install bridge-utils
```

手順

- (Ubuntu) Ubuntu 16.04.2 LTS で、以下のサードパーティ製パッケージがホストにインストールされていることを確認します。

```
dkms
ieee-data
javascript-common
libboost-chrono1.58.0
libboost-date-time1.58.0
libboost-program-options1.58.0
libgflags2v5
libgoogle-glog0v5
libgoogle-perftools4
libjs-jquery
libleveldb1v5
libprotobuf9v5
libyaml-0-2
python-gevent
python-linecache2
python-mako
python-netaddr
python-protobuf
python-simplejson
python-traceback2
python-unittest2
python-yaml
traceroute
```

依存関係のパッケージが Ubuntu 16.04.2 LTS にインストールされていない場合は、**apt-get install <package>** を実行してパッケージを手動でインストールします。

- (Red Hat) Red Hat 7.4 ホストが登録済みで、Red Hat リポジトリにアクセスできることを確認します。

注: NSX-T のユーザー インターフェイスを使用してホストを準備する場合、ホストに次の依存関係をインストールする必要があります。

- tcpdump
- boost-filesystem
- PyYAML

- boost-iostreams
 - boost-chrono
 - python-mako
 - python-netaddr
 - python-six
 - gperftools-lib
 - libunwind
 - snappy
 - boost-date-time
 - c-ares
 - redhat-lsb-core
 - wget
 - net-tools
 - yum-utils
- すでに Red Hat に登録されているホストを手動で準備する場合は、ホストに依存関係をインストールする必要はありません。Red Hat ホストが登録されていない場合は、**yum install <package>** を使用してリストにある依存関係を手動でインストールします。

RHEL KVM ホストの Open vSwitch のバージョンを確認する

OVS パッケージがホスト上にある場合は、既存のパッケージを削除し、サポートされているパッケージをインストールする必要があります。

Open vSwitch のサポート対象のバージョンは 2.9.1.8614397-1 です。

手順

- 1 ホストにインストールされている Open vSwitch の現在のバージョンを確認します。

ovs-vsitchd --version

サポートされるバージョンよりも新しい、または古い Open vSwitch を使用している場合は、Open vSwitch のサポート対象のバージョンに置き換える必要があります。

- a 次の Open vSwitch パッケージを削除します。

- kmod-openvswitch
- openvswitch
- openvswitch-selinux-policy

- b NSX-T を NSX Manger からインストールするか、手動のインストール手順を実行します。

2 または、NSX-T に必要な Open vSwitch パッケージをアップグレードします。

- a 管理者としてホストにログインします。
- b nsx-lcp ファイルをダウンロードし、/tmp ディレクトリにコピーします。
- c パッケージを解凍します。

```
tar -zxvf nsx-lcp-<release>-rhel73_x86_64.tar.gz
```

- d パッケージ ディレクトリに移動します。

```
cd nsx-lcp-rhel73_x86_64/
```

- e Open vSwitch の既存のバージョンをサポート対象のバージョンに置き換えます。

- 新しいバージョンの Open vSwitch が使用されている場合は、**--nodeps** コマンドを使用します。

例 : rpm-Uvh kmod-openvswitch-<new version>.e17.x86_64.rpm --nodeps

rpm-Uvh nicira-ovs-hypervisor-node-<new version>.x86_64.rpm --nodeps

rpm-Uvh openvswitch-*.rpm --nodeps

- Open vSwitch の古いバージョンが使用されている場合は、**--force** コマンドを使用します。

例 : rpm-Uvh kmod-openvswitch-<new version>.e17.x86_64.rpm --nodeps --force

rpm-Uvh nicira-ovs-hypervisor-node-<new version>.x86_64.rpm --nodeps --force

rpm-Uvh openvswitch-*.rpm --nodeps --force

次のステップ

NSX-T ファブリックへハイパーバイザー ホストを追加します。「[\[NSX-T ファブリックへのハイパーバイザー ホストの追加\]](#)」を参照してください。

NSX-T ファブリックへのハイパーバイザー ホストの追加

ファブリック ノードは、NSX-T 管理プレーンに登録されているノードであり、NSX-T のモジュールがインストールされています。ハイパーバイザー ホストを NSX-T オーバーレイの一部にするには、まず NSX-T ファブリックに追加する必要があります。

注: CLI を使用してモジュールを手動でホストにインストールし、ホストを管理プレーンに追加した場合は、この手順を省略できます。

注: RHEL 上の KVM ホストの場合は、**sudo** の認証情報を使用してホストの準備作業を実行できます。

前提条件

- NSX-T ファブリックに追加する各ホストについて、まず次のホスト情報を収集します。
 - ホスト名
 - 管理 IP アドレス

- ユーザー名
- パスワード
- (オプション) (KVM) SHA-256 SSL サムプリント
- (オプション) (ESXi) SHA-256 SSL サムプリント
- オプションで、ハイパーバイザー サムプリントを取得して、ホストをファブリックに追加するときに提供できるようにします。
- 情報を収集する方法の 1 つは、Linux シェルで次のコマンドを実行することです。

```
# echo -n | openssl s_client -connect <esxi-ip-address>:443 2>/dev/null | openssl x509
-noout -fingerprint -sha256
```

- ESX ホストで ESXi CLI を使用することもできます。

```
[root@host:~] openssl x509 -in /etc/vmware/ssl/rui.crt -fingerprint -sha256 -noout
SHA256 Fingerprint=49:73:F9:A6:0B:EA:51:2A:15:57:90:DE:C0:89:CA:7F:46:8E:30:15:CA:4D:
5C:95:28:0A:9E:A2:4E:3C:C4:F4
```

- KVM ハイパーバイザーから SHA-256 サムプリントを取得するには、KVM ホストでコマンドを実行します。

```
# awk '{print $2}' /etc/ssh/ssh_host_rsa_key.pub | base64 -d | sha256sum -b | sed
's/ .*$/ /' | xxd -r -p | base64
```

- Ubuntu の場合は、必須のサードパーティ製パッケージがインストールされていることを確認します。「[\[KVM ホストへのサードパーティ製パッケージのインストール\]](#)」を参照してください。

手順

- 1 NSX Manager の CLI で、install-upgrade サービスが実行されていることを確認します。

```
nsx-manager-1> get service install-upgrade

Service name: install-upgrade
Service state: running
Enabled: True
```

- 2 ブラウザから、NSX Manager (<https://<nsx-manager-ip-address>>) にログインします。
- 3 [ファブリック]-[ノード]-[ホスト] の順に選択し、[追加] をクリックします。

- 4 ホスト名、IP アドレス、ユーザー名、パスワードを入力します。サムプリントを入力することもできます。次はその例です。

ホストの追加



名前 *	comp-02b
IP アドレス *	<div>192.168.210.54 ×</div>
オペレーティング システム *	ESXi ▼
ユーザー名 *	root
パスワード *	●●●●●●
SHA-256 サムプリント	

キャンセル

追加

ホスト サムプリントを入力しない場合、NSX-T のユーザー インターフェイスで、ホストから取得したデフォルトのプレーン テキスト形式のサムプリントを使用するように求められます。

次はその例です。

無効なサムプリント



入力したサムプリントは無効です。

このサーバから提供されるサムプリントを使用しますか？

fa984ff00d4856c1e8db1be005ff908a3f2335bcd67776447e926aba71a006b8

いいえ

追加

ホストが正常に NSX-T ファブリックに追加されると、NSX Manager の [ファブリック] > [ノード] > [ホスト] で [展開ステータス：インストール成功] および [MPA 接続：アップ] が表示されます。ファブリック ノードをトランスポート ノードにするまで、[LCP 接続] は使用不可のままになります。

ホストを NSX-T ファブリックに追加した結果、NSX-T モジュールのコレクションがホストにインストールされます。ESXi では、モジュールが VIB としてパッケージングされます。RHEL 上の KVM の場合は、RPM としてパッケージングされます。Ubuntu 上の KVM の場合は、DEB としてパッケージングされます。

ESXi 上で確認するには、`esxcli software vib list | grep nsx` コマンドを実行します。この場合、インストールを実行した日が日付になります。

RHEL 上で確認するには、`yum list installed` または `rpm -qa` コマンドを実行します。

Ubuntu 上で確認するには、`dpkg --get-selections` コマンドを実行します。

ファブリック ノードは、GET `https://<nsx-mgr>/api/v1/fabric/nodes/<node-id>` API 呼び出しで確認できます。

```
{
  "resource_type" : "HostNode",
  "id" : "69b2a1d3-778d-4835-83c5-94cee99a213e",
  "display_name" : "10.143.1.218",
  "fqdn" : "w1-mvpccloud-218.eng.vmware.com",
  "ip_addresses" : [ "10.143.1.218" ],
  "external_id" : "69b2a1d3-778d-4835-83c5-94cee99a213e",
  "discovered_ip_addresses" : [ "10.143.1.218" ],
  "os_type" : "ESXI",
  "os_version" : "6.5.0",
  "managed_by_server" : "",
  "_create_user" : "admin",
  "_create_time" : 1498155416694,
  "_last_modified_user" : "admin",
```

```

"_last_modified_time" : 1498155416694,
"_protection" : "NOT_PROTECTED",
"_revision" : 0
}

```

API でのステータスは、GET <https://<nsx-mgr>/api/v1/fabric/nodes/<node-id>/status> API 呼び出しで監視できます。

```

{
  "lcp_connectivity_status" : "UP",
  "mpa_connectivity_status" : "UP",
  "last_sync_time" : 1480370899198,
  "mpa_connectivity_status_details" : "Client is responding to heartbeats",
  "lcp_connectivity_status_details" : [ {
    "control_node_ip" : "10.143.1.47",
    "status" : "UP"
  } ],
  "inventory_sync_paused" : false,
  "last_heartbeat_timestamp" : 1480369333415,
  "system_status" : {
    "mem_used" : 2577732,
    "system_time" : 1480370897000,
    "file_systems" : [ {
      "file_system" : "root",
      "total" : 32768,
      "used" : 5440,
      "type" : "ramdisk",
      "mount" : "/"
    }, {
      "file_system" : "etc",
      "total" : 28672,
      "used" : 264,
      "type" : "ramdisk",
      "mount" : "/etc"
    }, {
      "file_system" : "opt",
      "total" : 32768,
      "used" : 20,
      "type" : "ramdisk",
      "mount" : "/opt"
    }, {
      "file_system" : "var",
      "total" : 49152,
      "used" : 2812,
      "type" : "ramdisk",
      "mount" : "/var"
    }, {
      "file_system" : "tmp",
      "total" : 262144,
      "used" : 21728,
      "type" : "ramdisk",
      "mount" : "/tmp"
    }, {
      "file_system" : "iofilters",

```

```

    "total" : 32768,
    "used" : 0,
    "type" : "ramdisk",
    "mount" : "/var/run/iofilters"
  }, {
    "file_system" : "hostdstats",
    "total" : 116736,
    "used" : 2024,
    "type" : "ramdisk",
    "mount" : "/var/lib/vmware/hostd/stats"
  } ],
  "load_average" : [ 0.03999999910593033, 0.03999999910593033, 0.05000000074505806 ],
  "swap_total" : 0,
  "mem_cache" : 0,
  "cpu_cores" : 2,
  "source" : "cached",
  "mem_total" : 8386740,
  "swap_used" : 0,
  "uptime" : 3983605000
},
"software_version" : "2.0.0.0.4649755",
"host_node_deployment_status" : "INSTALL_SUCCESSFUL"
}

```

次のステップ

多数のハイパーバイザー（例：500 個以上）がある場合、NSX Manager で CPU の使用率が高くなり、パフォーマンスの問題が発生する可能性があります。この問題を回避するには、**aggsvc_change_intervals.py** スクリプトを実行します。このスクリプトは NSX ファイル ストアにあります（NSX CLI コマンド **copy file** または API **POST /api/v1/node/file-store/<file-name>?action=copy_to_remote_file** を使用して、ホストにスクリプトをコピーできます）。このスクリプトによって、特定のプロセスのポーリング間隔が変更されます。スクリプトを次のように実行します。

```
python aggsvc_change_intervals.py -m '<NSX Manager IP address>' -u 'admin' -p '<password>' -i 900
```

ポーリング間隔をデフォルト値に戻す場合は、次のようにします。

```
python aggsvc_change_intervals.py -m '<NSX Manager IP address>' -u 'admin' -p '<password>' -r
```

トランスポート ゾーンを作成します。「[「トランスポーティングゾーンについて」](#)」を参照してください。

NSX-T カーネル モジュールの手動インストール

NSX-T の [ファブリック] > [ノード] > [ホスト] > [追加] ユーザー インターフェイスまたは **POST /api/v1/fabric/nodes** API を使用する方法以外にも、NSX-T カーネル モジュールはハイパーバイザーのコマンドラインから手動でインストールすることもできます。

ESXi ハイパーバイザーへの NSX-T カーネル モジュールの手動インストール

ホストを NSX-T に追加するには、NSX-T カーネル モジュールを ESXi ホストにインストールする必要があります。インストールすると、NSX-T の制御プレーンと管理プレーン ファブリックを構築できます。VIB ファイルにパッケージされた NSX-T カーネル モジュールは、ハイパーバイザー カーネル内で実行され、分散ルーティング、分散ファイアウォール、ブリッジ機能などのサービスを提供します。

NSX-T の VIB を手動でダウンロードし、ホスト イメージに加えることができます。NSX-T の各リリースによって、ダウンロード パスが変わる場合があります。必ず NSX-T のダウンロード ページを確認し、適切な VIB を入手してください。

手順

- 1 root または管理者権限を持つユーザーでホストにログインします。
- 2 /tmp ディレクトリに移動します。

```
[root@host:~]: cd /tmp
```

- 3 nsx-lcp ファイルをダウンロードし、/tmp ディレクトリにコピーします。
- 4 インストール コマンドを実行します。

```
[root@host:/tmp]: esxcli software vib install -d /tmp/nsx-lcp-<release>.zip
Installation Result
  Message: Operation finished successfully.
  Reboot Required: false
  VIBs Installed: VMware_bootbank_nsx-aggservice_<release>, VMware_bootbank_nsx-da_<release>, VMware_bootbank_nsx-esx-datapath_<release>, VMware_bootbank_nsx-exporter_<release>, VMware_bootbank_nsx-host_<release>, VMware_bootbank_nsx-lldp_<release>, VMware_bootbank_nsx-mpa_<release>, VMware_bootbank_nsx-netcpa_<release>, VMware_bootbank_nsx-python-protobuf_<release>, VMware_bootbank_nsx-sfhc_<release>, VMware_bootbank_nsxa_<release>, VMware_bootbank_nsxcli_<release>
  VIBs Removed:
  VIBs Skipped:
```

ホストにインストール済みの要素に応じて、インストールされる VIB、削除される VIB、省略される VIB があります。コマンド出力に「**Reboot Required: true**」と表示されない限り、再起動は必要ありません。

ESXi ホストが NSX-T ファブリックに追加されると、次の VIB がホストにインストールされます。

- nsx-aggservice : NSX-T アグリゲーション サービスのホスト側ライブラリを提供します。NSX-T アグリゲーション サービスは、管理プレーン ノードで実行され、NSX-T コンポーネントからランタイム状態を取得するサービスです。
- nsx-da : 検出エージェント (DA) の、ハイパーバイザー OS バージョン、仮想マシン、ネットワーク インターフェイスに関するデータを収集します。データは管理プレーンに提供され、トラブルシューティング ツールで使用されます。
- nsx-esx-datapath : NSX-T のデータ プレーン パケット処理機能を提供します。

- `nsx-exporter` : 管理プレーンで実行されているアグリゲーション サービスにランタイム状態をレポートするホスト エージェントを提供します。
- `nsx-host` : ホストにインストールされている VIB バンドルにメタデータを提供します。
- `nsx-lldp` : Link Layer Discovery Protocol (LLDP) のサポートを提供します。LLDP は、ネットワーク デバイスで、その ID、機能、ネイバーを LAN でアドバタイズするために使用されるリンク レイヤー プロトコルです。
- `nsx-mpa` : NSX Manager とハイパーバイザー ホストの間の通信を提供します。
- `nsx-netcpa` : 中央の制御プレーンとハイパーバイザーの間の通信を提供します。中央の制御プレーンから論理ネットワークの状態を受け取り、この状態をデータ プレーンにプログラミングします。
- `nsx-python-protobuf` : プロトコル バッファに Python のバインドを提供します。
- `nsx-sfhc` : サービス ファブリック ホスト コンポーネント (SFHC) です。管理プレーンのインベントリでハイパーバイザーのライフサイクルをファブリック ホストとして管理するホスト エージェントを提供します。ハイパーバイザーにおける NSX-T のアップグレードやアンインストール、NSX-T モジュールの監視などの操作のチャネルとなります。
- `nsxa` : N-VDS の作成やアップリンクの設定など、ホストレベルの設定を行います。
- `nsxcli` : ハイパーバイザー ホストで NSX-T CLI を提供します。
- `nsx-support-bundle-client` : サポート バンドルを収集する機能を提供します。

確認するには、`[esxcli software vib list | grep nsx]` コマンドまたは `[esxcli software vib list | grep <yyyy-mm-dd>]` コマンドを ESXi ホストで実行します。日付は、インストールを行った日にします。

次のステップ

NSX-T の管理プレーンにホストを追加します。「[「ハイパーバイザー ホストの管理プレーンへの追加」](#)」を参照してください。

Ubuntu KVM ハイパーバイザーへの NSX-T カーネル モジュールの手動インストール

ホストを NSX-T に追加するときに、NSX-T カーネル モジュールを Ubuntu KVM ホストに手動でインストールできます。インストールすると、NSX-T の制御プレーンと管理プレーン ファブリックを構築できます。DEB ファイルにパッケージされた NSX-T カーネル モジュールは、ハイパーバイザー カーネル内で実行され、分散ルーティング、分散ファイアウォール、ブリッジ機能などのサービスを提供します。

NSX-T の DEB を手動でダウンロードし、ホスト イメージに加えることができます。ダウンロード パスは NSX-T のリリースごとに変わる可能性があるため、注意してください。必ず NSX-T のダウンロード ページを確認し、適切な DEB を入手してください。

前提条件

- 必要なサードパーティ製パッケージがインストールされていることを確認します。「[「KVM ホストへのサードパーティ製パッケージのインストール」](#)」を参照してください。

手順

- 1 管理者権限を持つユーザーでホストにログインします。

- (オプション) /tmp ディレクトリに移動します。

```
cd /tmp
```

- nsx-lcp ファイルをダウンロードし、/tmp ディレクトリにコピーします。
- パッケージを解凍します。

```
tar -xvf nsx-lcp-<release>-ubuntu-trusty-amd64.tar.gz
```

- パッケージディレクトリに移動します。

```
cd nsx-lcp-trusty-amd64/
```

- パッケージをインストールします。

```
sudo dpkg -i *.deb
```

- OVS カーネル モジュールを再読み込みします。

```
/etc/init.d/openvswitch-switch force-reload-kmod
```

ハイパーバイザーが OVS インターフェイスで DHCP を使用している場合は、DHCP が構成されているネットワーク インターフェイスを再起動します。ネットワーク インターフェイス上で古い dhclient プロセスを手動で停止し、そのインターフェイスで新しい dhclient プロセスを再開できます。

- 確認するには、`dpkg -l | grep nsx` コマンドを実行します。

```
user@host:~$ dpkg -l | grep nsx
```

ii	nsx-agent	<release>	amd64	NSX Agent
ii	nsx-aggservice	<release>	all	NSX Aggregation Service Lib
ii	nsx-cli	<release>	all	NSX CLI
ii	nsx-da	<release>	amd64	NSX Inventory Discovery Agent
ii	nsx-host	<release>	all	NSX host meta package
ii	nsx-host-node-status-reporter for Aggregation Service	<release>	amd64	NSX Host Status Reporter
ii	nsx-lldp	<release>	amd64	NSX LLDP Daemon
ii	nsx-logical-exporter	<release>	amd64	NSX Logical Exporter
ii	nsx-mpa	<release>	amd64	NSX Management Plane
	Agent Core			
ii	nsx-netcpa	<release>	amd64	NSX Netcpa
ii	nsx-sfhc	<release>	amd64	NSX Service Fabric
	Host Component			
ii	nsx-transport-node-status-reporter Status Reporter	<release>	amd64	NSX Transport Node
ii	nsxa	<release>	amd64	NSX L2 Agent

発生するほとんどのエラーは不完全な依存関係が原因です。`apt-get install -f` コマンドは、依存関係を解決し、NSX-T のインストールを再実行しようとします。

次のステップ

NSX-T の管理プレーンにホストを追加します。「[「ハイパーバイザー ホストの管理プレーンへの追加」](#)」を参照してください。

RHEL KVM ハイパーバイザーへの NSX-T カーネル モジュールの手動インストール

ホストを NSX-T に追加するときに、NSX-T カーネル モジュールを RHEL KVM ホストに手動でインストールできます。インストールすると、NSX-T の制御プレーンと管理プレーン ファブリックを構築できます。RPM ファイルにパッケージされた NSX-T カーネル モジュールは、ハイパーバイザー カーネル内で実行され、分散ルーティング、分散ファイアウォール、ブリッジ機能などのサービスを提供します。

NSX-T の RPM を手動でダウンロードし、ホスト イメージに加えることができます。ダウンロード パスは NSX-T のリリースごとに変わる可能性があるため、注意してください。必ず NSX-T のダウンロード ページを確認し、適切な RPM を入手してください。

前提条件

RHEL リポジトリにアクセスできること。

手順

- 1 管理者としてホストにログインします。
- 2 nsx-lcp ファイルをダウンロードし、/tmp ディレクトリにコピーします。
- 3 パッケージを解凍します。

```
tar -zxvf nsx-lcp-<release>-rhel73_x86_64.tar.gz
```

- 4 パッケージ ディレクトリに移動します。

```
cd nsx-lcp-rhel73_x86_64/
```

- 5 パッケージをインストールします。

```
sudo yum install *.rpm
```

yum インストール コマンドを実行すると、RHEL ホストが RHEL リポジトリにアクセスできる場合は、NSX-T の依存関係がすべて解決されます。

- 6 OVS カーネル モジュールを再読み込みします。

```
/etc/init.d/openvswitch force-reload-kmod
```

ハイパーバイザーが OVS インターフェイスで DHCP を使用している場合は、DHCP が構成されているネットワーク インターフェイスを再起動します。ネットワーク インターフェイス上で古い dhclient プロセスを手動で停止し、そのインターフェイスで新しい dhclient プロセスを再開できます。

- 7 確認するには、`rpm -qa | egrep 'nsx|openvswitch|nicira'` コマンドを実行します。

コマンドで表示されるインストールされたパッケージは、`nsx-rhel73` ディレクトリ内のパッケージと一致する必要があります。

次のステップ

NSX-T の管理プレーンにホストを追加します。「[「ハイパーバイザー ホストの管理プレーンへの追加」](#)」を参照してください。

ハイパーバイザー ホストの管理プレーンへの追加

ハイパーバイザー ホストを管理プレーンに追加すると、NSX Manager とホストが相互に通信できるようになります。

前提条件

NSX-T モジュールのインストールが完了している必要があります。

手順

- 1 NSX Manager アプライアンスへの SSH セッションを開きます。
- 2 管理者の認証情報を使用してログインします。
- 3 ハイパーバイザー ホストへの SSH セッションを開きます。
- 4 NSX Manager アプライアンスで、`get certificate api thumbprint cli` コマンドを実行します。

コマンド出力は、この NSX Manager に固有の一連の数値です。

次はその例です。

```
NSX-Manager1> get certificate api thumbprint
...
```

- 5 ハイパーバイザー ホストで、`[nsxcli]` コマンドを実行して、NSX-T CLI に入ります。

注: KVM の場合はスーパーユーザー (`sudo`) でコマンドを実行します。

```
[user@host:~] nsxcli
host>
```

プロンプトが変わります。

- 6 ハイパーバイザー ホストで `[join management-plane]` コマンドを実行します。

このとき、次の情報を指定します。

- NSX Manager のホスト名または IP アドレスとオプションでポート番号
- NSX Manager のユーザー名
- NSX Manager の証明書サムプリント

■ NSX Manager のパスワード

```
host> join management-plane NSX-Manager1 username admin thumbprint <NSX-Manager1's-thumbprint>
Password for API user: <NSX-Manager1's-password>
Node successfully joined
```

ホストで **get managers** コマンドを実行して結果を確認します。

```
host> get managers
- 192.168.110.47    Connected
```

NSX Manager のユーザー インターフェイスの [ファブリック] > [ノード] > [ホスト] で、ホストの MPA 接続が [稼働中] になっていることを確認します。

ファブリック ホストの状態は [GET https://<nsx-mgr>/api/v1/fabric/nodes/<fabric-node-id>/state] API 呼び出しで確認することもできます。

```
{
  "details": [],
  "state": "success"
}
```

管理プレーンからホストの証明書が制御プレーンに送信され、制御プレーンによって制御プレーンの情報がホストにプッシュされます。

各 ESXi ホストの `/etc/vmware/nsx/controller-info.xml` に NSX Controller のアドレスが記載されています。または **get controllers** を使用して CLI にアクセスしてください。

```
[root@host:~] cat /etc/vmware/nsx/controller-info.xml
<?xml version="1.0" encoding="utf-8"?>
<config>
  <connectionList>
    <connection id="0">
      <server>10.143.1.47</server>
      <port>1234</port>
      <sslEnabled>true</sslEnabled>
      <pemKey>-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----</pemKey>
    </connection>
    <connection id="1">
      <server>10.143.1.45</server>
      <port>1234</port>
      <sslEnabled>true</sslEnabled>
      <pemKey>-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----</pemKey>
    </connection>
    <connection id="2">
      <server>10.143.1.46</server>
      <port>1234</port>
      <sslEnabled>true</sslEnabled>
```

```

    <pemKey>-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----</pemKey>
  </connection>
</connectionList>
</config>

```

NSX-T へのホスト接続が開始され、ホストがトランスポート ノードに昇格するまで「CLOSE_WAIT」ステータスで維持されます。これは [esxcli network ip connection list | grep 1234] コマンドで確認できます。

```

# esxcli network ip connection list | grep 1234
tcp          0      0 192.168.210.53:45823      192.168.110.34:1234  [CLOSE_WAIT]    37256
newreno      netcpa

```

KVM の場合は `netstat -anp --tcp | grep 1234` コマンドを使用します。

```

user@host:~$ netstat -anp --tcp | grep 1234
tcp  0      0 192.168.210.54:57794      192.168.110.34:1234    [CLOSE_WAIT] -

```

次のステップ

トランスポート ゾーンを作成します。「[「トランスポートゾーンについて」](#)」を参照してください。

トランスポート ゾーンとトランスポート ノード

トランスポート ゾーンとトランスポート ノードは、NSX-T における重要な概念です。

この章には、次のトピックが含まれています。

- トランストランスポート ゾーンについて
- 拡張データパス
- トンネル エンドポイントの IP アドレス用 IP アドレス プールの作成
- アップリンク プロファイルの作成
- トランスポート ゾーンの作成
- ホスト トランスポート ノードの作成
- Network I/O Control の設定
- NSX Edge トランスポート ノードの作成
- NSX Edge クラスタの作成

トランストランスポート ゾーンについて

トランスポート ゾーンは、トランスポート ノードのおよぶ範囲を定義するコンテナです。トランスポート ノードはハイパーバイザー ホストおよび NSX Edge で、NSX-T オーバーレイに参加します。ハイパーバイザー ホストの場合は、NSX-T 論理スイッチを介して通信する仮想マシンをホストします。NSX Edge の場合は、論理ルーターのアップリンクとダウンリンクを持ちます。

トランスポート ゾーンを作成するときは、N-VDS モードを [標準 (Standard)] または [拡張データパス (Enhanced Datapath)] のいずれかに指定する必要があります。トランスポート ゾーンにトランスポート ノードを追加すると、トランスポート ゾーンに関連付けられている N-VDS がトランスポート ノードにインストールされます。各トランスポート ゾーンは、1 台の N-VDS をサポートします。拡張データパス N-VDS は、NFV (ネットワーク機能の仮想化) のワークロードをサポートするパフォーマンス機能を持ち、VLAN とオーバーレイ ネットワークの両方をサポートし、拡張データパス N-VDS をサポートする ESXi ホストが必要です。

トランスポート ノードは、以下に属することが可能です。

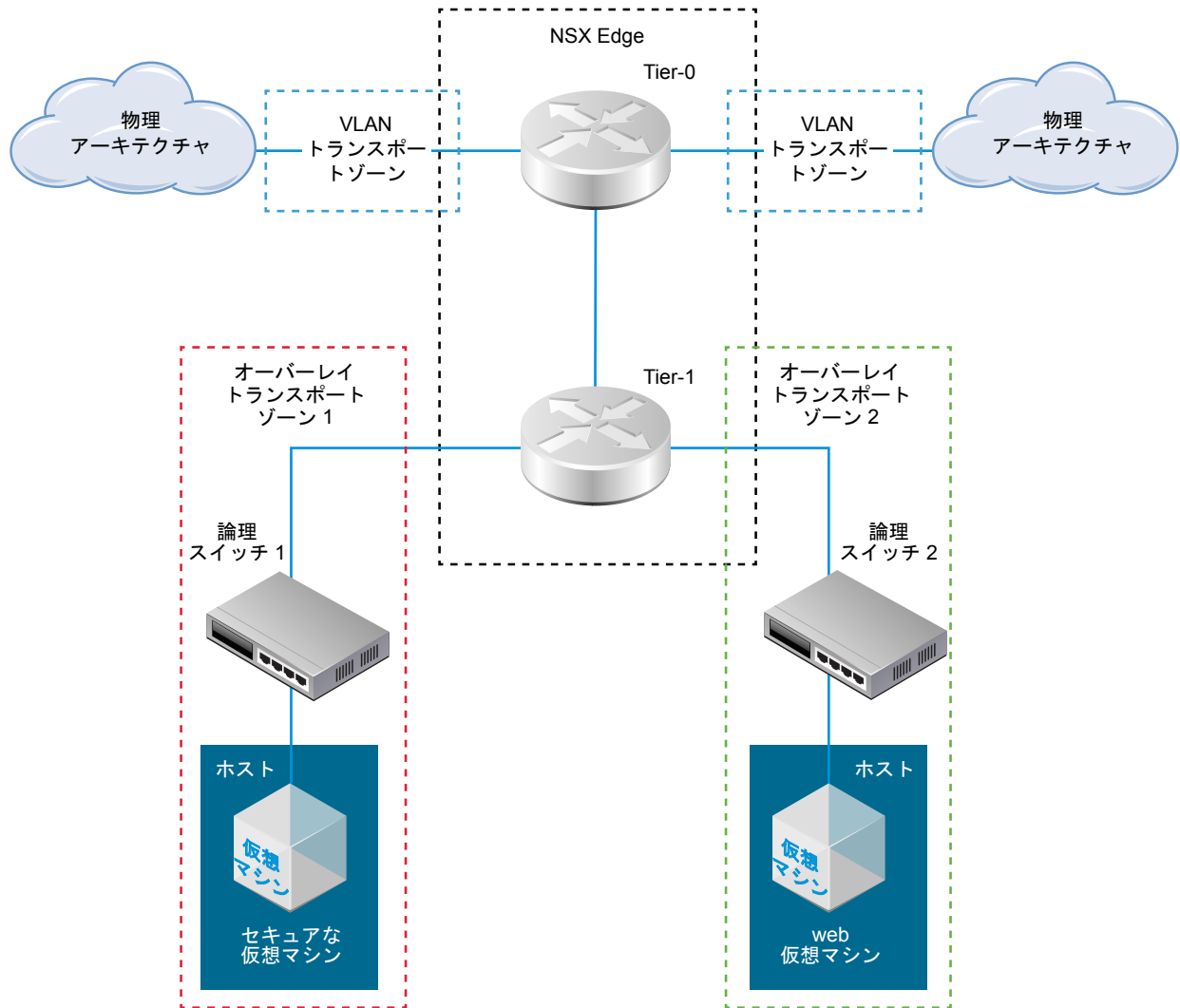
- 複数の VLAN トランスポート ゾーン。
- 標準の N-VDS を持つ、最大 1 つのオーバーレイ トランスポート ゾーン。
- トランスポート ノードが ESXi ホスト上で実行されている場合、高度なデータパスの N-VDS が配置された複数のオーバーレイ トランスポート ゾーン。

2 台のトランスポート ノードが同じトランスポート ゾーンにある場合、両方のトランスポート ノードでホストされる仮想マシンは、同じトランスポート ゾーン内の NSX-T 論理スイッチに接続できます。これにより、仮想マシンがレイヤー 2/レイヤー 3 に到達できる場合は、仮想マシン同士が互いに通信できるようになります。各仮想マシンが、それぞれ別のトランスポート ゾーン内のスイッチに接続されている場合、それらの仮想マシンは互いに通信できません。トランスポート ゾーンは、レイヤー 2/レイヤー 3 の接続性要件に変わるものではありませんが、接続性に制約を加えます。つまり、相互に接続するには、前提条件として同じトランスポート ゾーンに属する必要があります。この前提条件が満たされれば、相互接続は可能になりますが、自動的に通信が可能となるわけではありません。実際に接続を可能にするには、レイヤー 2 および別のサブネットの場合のレイヤー 3 のネットワークの設定と条件が正しく動作している必要があります。

1 個のトランスポート ノードに通常の仮想マシンと高セキュリティ仮想マシンの両方が含まれているとします。使用するネットワーク設計では、通常の仮想マシンは互いに到達可能ですが、セキュリティ仮想マシンには到達できません。これを実現するには、`secure-tz` という名前の 1 つのトランスポート ゾーンに属するホスト上に、セキュアな仮想マシンを配置します。通常のセキュアな仮想マシンは、同じトランスポート ノードに配置することはできません。通常の仮想マシンは、`general-tz` という名前の別のトランスポート ゾーンに配置します。通常の仮想マシンは、`general-tz` に置かれた NSX-T 論理スイッチに接続されます。高セキュリティ仮想マシンは、`secure-tz` に置かれた NSX-T 論理スイッチに接続されます。異なるトランスポート ゾーンに置かれた仮想マシン同士は、同じサブネットにあっても、互いに通信できません。仮想マシンから論理スイッチへの接続によって、最終的に仮想マシンの到達可能性が制御されます。このように、2 台の論理スイッチが別のトランスポート ゾーンに置かれているため、Web 仮想マシンとセキュア仮想マシンは互いに通信できません。

たとえば、次の図は、3 つのトランスポート ゾーン (2 つの VLAN トランスポート ゾーンとオーバーレイ トランスポート ゾーン 2) に属する NSX Edge を示しています。オーバーレイ トランスポート ゾーン 1 には、1 台のホスト、1 台の NSX-T 論理スイッチ、および 1 台のセキュア仮想マシンが含まれています。NSX Edge はオーバーレイ トランスポート ゾーン 1 に属さないため、セキュア仮想マシンは物理アーキテクチャにアクセスできません。これに対して、オーバーレイ トランスポート ゾーン 2 内の Web 仮想マシンは物理アーキテクチャと通信できます。これは、NSX Edge がオーバーレイ トランスポート 2 に属するためです。

図 8-1. NSX-T トランスポート ゾーン



拡張データ パス

拡張データ パスはネットワーク スタック モードであり、これを構成することで優れたネットワーク パフォーマンスを実現します。これは主に NFV ワークロードを対象としています。NFV ワークロードでは、このモードで実現するパフォーマンス上の利点を活用する必要があります。

N-VDS スイッチは、ESXi ホスト上でのみ拡張データ パス モードで設定できます。

拡張データ パス モードでは、以下を設定できます。

- VLAN トラフィック

拡張データ パスを構成する手順の概要

ネットワーク管理者は、拡張データ パス モードで N-VDS をサポートするトランスポート ゾーンを作成する前に、サポートされている NIC カードおよびドライバを使用してネットワークを準備する必要があります。ネットワークパフォーマンスを向上させるために、ロード バランシングされた送信元チーミング ポリシーを有効にして、NUMA ノードを認識させることができます。

手順の概要は次のとおりです。

- 1 拡張データ パスをサポートする NIC カードを使用します。

拡張データ パスをサポートする NIC カードについては、[VMware 互換性ガイド](#)を参照してください。

VMware 互換性ガイドのページの [I/O デバイス] カテゴリで、[ESXi 6.7]、I/O デバイスのタイプに [ネットワーク]、機能に [N-VDS 拡張データパス] を選択します。

- 2 [My VMware ページ](#) から NIC ドライバをダウンロードしてインストールします。

- 3 アップリンク ポリシーを作成します。

「[アップリンク プロファイルの作成](#)」を参照してください。

- 4 N-VDS を持つトランスポート ゾーンを拡張データ パス モードで作成します。

「[トランスポート ゾーンの作成](#)」を参照してください。

- 5 ホスト トランスポート ノードを作成します。論理コアと NUMA ノードを持つ拡張データ パス N-VDS を設定します。

「[ホスト トランスポート ノードの作成](#)」を参照してください。

トンネル エンドポイントの IP アドレス用 IP アドレス プールの作成

トンネル エンドポイント用に IP アドレス プールを使用できます。トンネル エンドポイントは、NSX-T でカプセル化されたオーバーレイ フレームの送信と終了を行うハイパーバイザー ホストを一意に識別するために外部 IP ヘッダで使用される、送信先 IP アドレスと宛先 IP アドレスです。トンネル エンドポイントの IP アドレスには、DHCP または手動で設定した IP アドレス プールも使用できます。

ESXi ホストと KVM ホストの両方を使用している場合、設計オプションの 1 つとして、ESXi トンネル エンドポイント IP アドレス プール (sub_a) と KVM トンネル エンドポイント IP アドレス プール (sub_b) 用に 2 つの異なるサブ ネットを使用できます。この場合、専用のデフォルト ゲートウェイを使用して、KVM ホスト上で sub_a へのスタティック ルートを追加する必要があります。

次の例は、sub_a が 192.168.140.0 で sub_b が 192.168.150.0 の、Ubuntu ホスト上のルーティング テーブルを示しています (たとえば、管理サブネットは 192.168.130.0 になります)。

カーネル IP アドレス ルーティング テーブル：

Destination	Gateway	Genmask	Iface
0.0.0.0	192.168.130.1	0.0.0.0	eth0
192.168.122.0	0.0.0.0	255.255.255.0	virbr0
192.168.130.0	0.0.0.0	255.255.255.0	eth0
192.168.140.0	192.168.150.1	255.255.255.0	nsx-vtep0.0
192.168.150.0	0.0.0.0	255.255.255.0	nsx-vtep0.0

root は複数の方法で追加できます。たとえば次の 2 つの方法があります。2 つの方法のうち、インターフェイスを編集してルートを追加した場合のみ、ホストの再起動後もルートが維持されます。route add コマンドを使用して追加したルートは、ホストの再起動後は維持されません。

```
route add -net 192.168.140.0 netmask 255.255.255.0 gw 192.168.150.1 dev nsx-vtep0.0
```

/etc/network/interfaces の「up ifconfig nsx-vtep0.0 up」の前に、このスタティック ルートを追加します。

```
post-up route add -net 192.168.140.0 netmask 255.255.255.0 gw 192.168.150.1
```

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [インベントリ] > [グループ] > [IP アドレス プール] の順に選択し、[追加] をクリックします。
- 3 IP アドレス プールの名前、説明（オプション）、ネットワーク設定を入力します。

ネットワーク設定には次のものが含まれます。

- IP アドレスの範囲
- ゲートウェイ
- CIDR 形式のネットワーク アドレス
- （オプション）DNS サーバのコンマ区切りのリスト

■ (オプション) DNS サフィックス

次はその例です。

新しい IP アドレス プールの追加



名前 *	corp-tep
説明	

サブネット

+ 追加 削除

<input checked="" type="checkbox"/> IP アドレス範囲 *	ゲートウェイ	CIDR *	DNS サーバ	DNS サフィックス
<input checked="" type="checkbox"/> 192.168.250.100 - 192.168.250.200	192.168.210.1	192.168.250.0/24		corp.local

キャンセル

追加

IP アドレス プールは、GET <https://<nsx-mgr>/api/v1/pools/ip-pools> API 呼び出しでも確認できます。

```
{
  "cursor": "0036e2d8c2e8-f6d7-498e-821b-b7e44d2650a9ip-pool-1",
  "sort_by": "displayName",
  "sort_ascending": true,
  "result_count": 1,
  "results": [
    {
      "id": "e2d8c2e8-f6d7-498e-821b-b7e44d2650a9",
      "display_name": "comp-tep",
      "resource_type": "IpPool",
      "subnets": [
        {
          "dns_nameservers": [
            "192.168.110.10"
          ],
          "allocation_ranges": [
            {
              "start": "192.168.250.100",
              "end": "192.168.250.200"
            }
          ],
          "gateway_ip": "192.168.250.1",
          "cidr": "192.168.250.0/24",
          "dns_suffix": "corp.local"
        }
      ]
    }
  ]
}
```

```

    ],
    "_last_modified_user": "admin",
    "_last_modified_time": 1443649891178,
    "_create_time": 1443649891178,
    "_system_owned": false,
    "_create_user": "admin",
    "_revision": 0
  }
]
}

```

次のステップ

アップリンク プロファイルを作成します。「[アップリンク プロファイルの作成](#)」を参照してください。

アップリンク プロファイルの作成

アップリンク プロファイルでは、ハイパーバイザー ホストから NSX-T 論理スイッチまたは NSX Edge ノードから トップオブラック スイッチへのリンクのポリシーを定義します。

アップリンク プロファイルでは、チーミング ポリシー、アクティブ/スタンバイ リンク、トランスポート VLAN ID、MTU 設定などを定義します。

アップリンク プロファイルを使用することで、複数のホストやノードのネットワーク アダプタに同じ機能を設定できます。アップリンク プロファイルは、ネットワーク アダプタに設定するプロパティや機能のコンテナです。ネットワーク アダプタごとに個別にプロパティや機能を設定するのではなく、アップリンク プロファイルで機能を指定できます。これは、NSX-T のトランスポート ノードを作成するときに適用できます。

仮想マシン/アプライアンス ベースの NSX Edge では、スタンバイ アップリンクはサポートされません。仮想アプライアンスとして NSX Edge をインストールする場合は、デフォルトのアップリンク プロファイルを使用します。仮想マシン ベースの NSX Edge 向けに作成された各アップリンク プロファイルは、1 つのアクティブ アップリンクを指定し、スタンバイ アップリンクは指定しません。

注: 異なる VLAN を使用してアップリンクごとに別の N-VDS を作成する場合は、NSX Edge 仮想マシンで複数のアップリンクを設定できます。アップリンクごとに個別の VLAN トランスポート ゾーンが必要です。これは、複数の ToR スイッチに接続する単一の NSX Edge ノードをサポートするための機能です。

前提条件

- NSX Edge ネットワークについて理解しておく必要があります。「[NSX Edge のネットワーク設定](#)」を参照してください。
- アップリンク プロファイル内の各アップリンクは、ハイパーバイザー ホストまたは NSX Edge ノードの、稼動中で使用可能な物理リンクに対応している必要があります。

たとえば、ハイパーバイザー ホストで稼働中の物理リンクとして、vmnic0 と vmnic1 の 2 つがあるとします。vmnic0 は管理ネットワークとストレージ ネットワークに使用され、vmnic1 は未使用であるとします。この場合、vmnic1 を NSX-T のアップリンクとして使用できますが、vmnic0 は使用できません。リンクのチーミングを行うには、vmnic1 と vmnic2 など、未使用の物理リンクが 2 つ必要です。

NSX Edge については、トンネル エンドポイントと VLAN アップリンクに同じ物理リンクを使用できます。たとえば、vmnic0/eth0/em0 を管理ネットワークに使用し、vmnic1/eth1/em1 を fp-ethX リンクに使用することが可能です。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック]-[プロファイル]-[アップリンク プロファイル] の順に選択し、[追加] をクリックします。
- 3 アップリンク プロファイルの詳細を入力します。

オプション	説明
名前	アップリンク プロファイルの名前を入力します。
説明	オプションで、アップリンク プロファイルの説明を追加します。

オプション	説明
LAG	<p>(オプション) トランスポート ネットワークに Link Aggregation Control Protocol (LACP) を使用するリンク アグリゲーション グループ (LAG)。</p> <p>注: LACP の場合、複数の LAG は KVM ホストでサポートされません。</p> <p>アクティブなアップリンクの名前を含むカンマ区切りのリストを追加します。</p> <p>スタンバイ アップリンクの名前を含むカンマ区切りのリストを追加します。作成するアクティブ アップリンクとスタンバイ アップリンクの名前には、物理リンクを表す任意のテキストを指定できます。これらのアップリンク名は、後でトランスポート ノードを作成するときに参照します。トランスポート ノードのユーザー インターフェイス/API で、各アップリンク名に対応する物理リンクを指定できます。</p>
チーミング	<p>[チーミング] セクションで [追加] をクリックして、詳細を入力します。チーミング ポリシーは、N-VDS で冗長性とトラフィックのロード バランシングのためにアップリンクをどのように使用するかを定義します。チーミング ポリシーは、2 つのチーミング ポリシー モードで設定できます。</p> <ul style="list-style-type: none"> ■ [フェイルオーバー順序]: 1 つのアクティブ アップリンクと、オプションでスタンバイ アップリンクのリストを指定します。アクティブ アップリンクに障害が発生した場合、スタンバイ リスト内の次のアップリンクで置き換えられます。このオプションでは、ロード バランシングは実際には実行されません。 ■ [ロード バランシング ソース]: アクティブ アップリンクのリストを指定し、トランスポート ノード上の各インターフェイスを 1 つのアクティブ アップリンクに固定します。この構成では、同時に複数のアクティブ アップリンクを使用できます。 <p>注: KVM ホストでサポートされるのは、フェイルオーバーの順序のチーミング ポリシーのみです。ロード バランシングの送信元のチーミング ポリシーはサポートされていません。</p> <p>(ESXi ホストのみ) トランスポート ゾーンに次のポリシーを定義できます。</p> <ul style="list-style-type: none"> ■ スイッチに設定された論理スイッチごとの名前付きチーミング ポリシー。 ■ スイッチ全体のデフォルトのチーミング ポリシー。 <p>名前付きのチーミング ポリシー: 名前付きのチーミング ポリシーを使用すると、論理スイッチごとに特定のチーミング ポリシー モードおよびアップリンクを定義できます。このポリシータイプにより、バンド幅の要件に応じてアップリンクを柔軟に選択できるようになります。</p> <ul style="list-style-type: none"> ■ 名前付きのチーミング ポリシーを定義すると、N-VDS は、接続されているトランスポート ゾーンおよびホスト内の論理スイッチによって指定された場合に、その名前付きチーミング ポリシーを使用します。 ■ 名前付きのチーミング ポリシーを定義しない場合、N-VDS はデフォルトのチーミング ポリシーを使用します。

4 トランスポート VLAN の値を入力します。

5 MTU 値を入力します。

デフォルト値は 1600 です。

ユーザー インターフェイスに加え、GET /api/v1/host-switch-profiles API 呼び出しでアップリンク プロファイルを表示することもできます。

```
{
  "result_count": 2,
  "results": [
    {
      "resource_type": "UplinkHostSwitchProfile",
      "id": "16146a24-122b-4274-b5dd-98b635e4d52d",

```

```

    "display_name": "comp-uplink",
    "transport_vlan": 250,
    "teaming": {
      "active_list": [
        {
          "uplink_type": "PNIC",
          "uplink_name": "uplink-1"
        }
      ],
      "standby_list": [ {
        "uplink_name": "uplink-2",
        "uplink_type": "PNIC"
      } ],
      "policy": "FAILOVER_ORDER"
    },
    "mtu": 1600,
    "_last_modified_time": 1457984399526,
    "_create_time": 1457984399526,
    "_last_modified_user": "admin",
    "_system_owned": false,
    "_create_user": "admin",
    "_revision": 0
  },
  {
    "resource_type": "UplinkHostSwitchProfile",
    "id": "c9e35cec-e9d9-4c51-b52e-17a5c1bd9a38",
    "display_name": "vlan-uplink",
    "transport_vlan": 100,
    "teaming": {
      "active_list": [
        {
          "uplink_type": "PNIC",
          "uplink_name": "uplink-1"
        }
      ],
      "standby_list": [],
      "policy": "FAILOVER_ORDER"
    },
    "named_teamings": [
      {
        "active_list": [
          {
            "uplink_type": "PNIC",
            "uplink_name": "uplink-2"
          }
        ],
        "standby_list": [
          {
            "uplink_type": "PNIC",
            "uplink_name": "uplink-1"
          }
        ],
        "policy": "FAILOVER_ORDER",
        "name": "named teaming policy"
      }
    ]
  }

```

```

    ]
    "mtu": 1600,
    "_last_modified_time": 1457984399574,
    "_create_time": 1457984399574,
    "_last_modified_user": "admin",
    "_system_owned": false,
    "_create_user": "admin",
    "_revision": 0
  }
]
}

```

次のステップ

トランスポート ゾーンを作成します。「[「トランスポート ゾーン」の作成](#)」を参照してください。

トランスポート ザーンの作成

トランスポート ザーンでは、特定のネットワークを使用できるホストと仮想マシンを指定します。トランスポート ザーンで論理スイッチを認識できるホストを制限し、論理スイッチに接続できる仮想マシンを制限することで、この制御が実現します。1 つのトランスポート ザーンの範囲が、1 つ以上のクラスタにまたがることができます。

NSX-T 環境には、要件に基づいて 1 つ以上のトランスポート ザーンを含めることができます。1 台のホストが、複数のトランスポート ザーンに属することができます。論理スイッチは、1 つのトランスポート ザーンのみ属することができます。

NSX-T では、レイヤー 2 ネットワークの異なるトランスポート ザーンにある仮想マシンに接続できません。論理スイッチの範囲は 1 つのトランスポート ザーンに制限されるため、異なるトランスポート ザーンにある仮想マシンは同じレイヤー 2 ネットワーク上に配置できません。

オーバーレイ トランスポート ザーンは、ホストトランスポート ノードと NSX Edge の両方で使用されます。ホストまたは NSX Edge トランスポート ノードがオーバーレイ トランスポート ザーンに追加されると、N-VDS がそのホストまたは NSX Edge にインストールされます。

VLAN トランスポート ザーンは、NSX Edge が VLAN アップリンクに使用します。NSX Edge が VLAN トランスポート ザーンに追加されると、VLAN N-VDS が NSX Edge にインストールされます。

N-VDS によって論理ルーター アップリンクおよびダウンリンクが物理 NIC にバインドされることで、仮想から物理へのパケット フローが可能になります。

トランスポート ザーンを作成する際には、そのトランスポート ザーンに後で追加されるトランスポート ノードにインストールされる、N-VDS の名前を指定する必要があります。N-VDS には任意の名前を付けることができます。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック]-[トランスポート ザーン]-[追加] を選択します。
- 3 トランスポート ザーンの名前と、必要に応じて説明を入力します。
- 4 N-VDS の名前を入力します。

- 5 N-VDS モードを選択します。

オプションは [標準] と [拡張データパス] です。

- 6 N-VDS モードが [標準] の場合は、トラフィック タイプを選択します。

オプションは、[オーバーレイ] と [VLAN] です。

- 7 N-VDS モードが [拡張データパス] の場合は、トラフィック タイプを選択します。

オプションは、[オーバーレイ] と [VLAN] です。

注: [拡張データパス] モードでは、特定の NIC 構成のみがサポートされます。サポート対象の NIC を構成していることを確認します。

- 8 1 つ以上のアップリンク チーミング ポリシー名を入力します。これらの名前付きチーミング ポリシーは、トランスポート ゾーンに接続される論理スイッチで使用できます。論理スイッチで一致する名前付きチーミング ポリシーが見つからない場合は、デフォルトのアップリンク チーミング ポリシーが使用されます。

- 9 [トランスポート ゾーン] ページで、新規トランスポート ゾーンを確認します。

- 10 (オプション) 新しいトランスポートゾーンは、GET <https://<nsx-mgr>/api/v1/transport-zones> API 呼び出しで確認することもできます。

```
{
  "cursor": "00369b661aed-1eaa-4567-9408-ccbcfe50b416tz-vlan",
  "result_count": 2,
  "results": [
    {
      "resource_type": "TransportZone",
      "description": "comp overlay transport zone",
      "id": "efd7f38f-c5da-437d-af03-ac598f82a9ec",
      "display_name": "tz-overlay",
      "host_switch_name": "overlay-hostswitch",
      "transport_type": "OVERLAY",
      "transport_zone_profile_ids": [
        {
          "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
          "resource_type": "BfdHealthMonitoringProfile"
        }
      ],
      "_create_time": 1459547126454,
      "_last_modified_user": "admin",
      "_system_owned": false,
      "_last_modified_time": 1459547126454,
      "_create_user": "admin",
      "_revision": 0,
      "_schema": "/v1/schema/TransportZone"
    },
    {
      "resource_type": "TransportZone",
      "description": "comp vlan transport zone",
      "id": "9b661aed-1eaa-4567-9408-ccbcfe50b416",
      "display_name": "tz-vlan",
```

```

    "host_switch_name": "vlan-uplink-hostswitch",
    "transport_type": "VLAN",
    "transport_zone_profile_ids": [
      {
        "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
        "resource_type": "BfdHealthMonitoringProfile"
      }
    ],
    "_create_time": 1459547126505,
    "_last_modified_user": "admin",
    "_system_owned": false,
    "_last_modified_time": 1459547126505,
    "_create_user": "admin",
    "_revision": 0,
    "_schema": "/v1/schema/TransportZone"
  }
]
}

```

次のステップ

オプションで、カスタム トランスポート ゾーン プロファイルを作成し、それをトランスポート ゾーンにバインドします。カスタム トランスポート ゾーン プロファイルは、**POST /api/v1/transportzone-profiles** API を使用して作成できます。トランスポート ゾーン プロファイルの作成にユーザー インターフェイスを使用するワークフローはありません。トランスポート ゾーン プロファイルの作成後は、**PUT /api/v1/transport-zones/<transport-zone-id>** API を使用してトランスポート ゾーンで確認できます。

トランスポート ノードを作成します。「[\[「ホスト トランスポート ノードの作成」\]](#)」を参照してください。

ホスト トランスポート ノードの作成

トランスポート ノードは、NSX-T オーバーレイまたは NSX-T VLAN ネットワークに参加するノードです。

KVM ホストの場合は、N-VDS を事前に設定できます。また、NSX Manager で設定を実行することも可能です。ESXi ホストの場合は、常に NSX Manager で N-VDS が設定されます。

注: テンプレート仮想マシンを使用してトランスポート ノードを作成する場合は、ホストの `/etc/vmware/nsx/` に証明書がないことを確認してください。証明書がすでに存在する場合、netcpa エージェントは証明書を作成しません。

前提条件

- ホストが管理プレーンに追加され、[ファブリック] > [ホスト] ページで MPA 接続が確立されている必要があります。
- トランスポート ゾーンが設定されている必要があります。
- アップリンク プロファイルが設定されている必要があります。設定されていない場合は、デフォルトのアップリンク プロファイルを使用できます。
- IP アドレス プールが設定されているか、ネットワーク環境 DHCP が使用できる必要があります。

- ホスト上で 1 個以上の未使用の物理 NIC が必要です。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック] - [ノード] - [トランスポート ノード] - [追加] の順に選択します。
- 3 トランスポート ノードの名前を入力します。
- 4 ドロップダウン メニューからノードを選択します。
- 5 このトランスポート ノードが属するトランスポート ゾーンを選択します。
- 6 [N-VDS] タブをクリックします。
- 7 KVM ノードの場合は、N-VDS タイプを選択します。

オプション	説明
標準	NSX Manager は N-VDS を作成します。 このオプションはデフォルトで選択されています。
事前設定済み	N-VDS はすでに設定されています。

非 KVM ノードの場合、N-VDS タイプは常に [標準] または [拡張データパス] です。

- 8 標準の N-VDS の場合は、次の詳細を提供します。

オプション	説明
N-VDS 名	このノードが属するトランスポート ゾーンの N-VDS 名と同じにする必要があります。
NIOC プロファイル	ドロップダウン メニューから NIOC プロファイルを選択します。
アップリンク プロファイル	ドロップダウン メニューからアップリンク プロファイルを選択します。
IP アドレスの割り当て	[DHCP を使用]、[IP アドレス プールを使用] または [固定 IP リストを使用] を選択します。 [固定 IP アドレスのリストを使用] を選択した場合は、IP アドレス、ゲートウェイ、およびサブネット マスクのコンマ区切りのリストを指定する必要があります。
IP アドレス プール	IP 割り当てに [IP アドレス プールを使用] を選択した場合は、IP アドレス プール名を指定します。
物理 NIC	物理 NIC が使用されていないことを確認します。標準の vSwitch または vSphere Distributed Switch などが使用していることがあります。すでに使用されている場合、トランスポート ノードは [部分的成功] 状態になり、ファブリック ノードの LCP 接続の確立に失敗します。

- 9 拡張データパス N-VDS の場合は、次の詳細を指定します。

オプション	説明
N-VDS 名	このノードが属するトランスポート ゾーンの N-VDS 名と同じにする必要があります。
IP アドレスの割り当て	[DHCP を使用]、[IP アドレス プールを使用] または [固定 IP リストを使用] を選択します。 [固定 IP アドレスのリストを使用] を選択した場合は、IP アドレス、ゲートウェイ、およびサブネット マスクのコンマ区切りのリストを指定する必要があります。
IP アドレス プール	IP アドレスの割り当てに [IP アドレス プールを使用] を選択した場合は、IP アドレス プール名を指定します。

オプション	説明
物理 NIC	拡張データベース対応の物理 NIC を選択します。物理 NIC が使用されていないことを確認します。標準の vSwitch または vSphere Distributed Switch などが使用していることがあります。すでに使用されている場合、トランスポート ノードは [部分的成功] 状態になり、ファブリック ノードの LCP 接続の確立に失敗します。
アップリンク	ドロップダウン メニューからアップリンク プロファイルを選択します。
CPU の設定	<p>[NUMA ノード インデックス] ドロップダウン メニューで、N-VDS スイッチに割り当てる NUMA ノードを選択します。ノード上にある最初の NUMA ノードは、値 0 で表されます。</p> <p>esxcli hardware memory get コマンドを実行して、ホスト上の NUMA ノード数を確認できます。</p> <p>注: N-VDS スイッチとアフィニティがある NUMA ノードの数を変更する場合は、NUMA ノード インデックス値を更新することができます。</p> <p>[NUMA ノードあたりの Lcore 数] ドロップダウン メニューで、拡張データベースで使用する必要がある論理コアの数を選択します。</p> <p>esxcli network ens maxLcores get コマンドを実行して、NUMA ノード上に作成できる論理コアの最大数を確認できます。</p> <p>注: 使用可能な NUMA ノードと論理コアがすべて使用されている場合、トランスポート ノードに追加した新しいスイッチは ENS トラフィック用に有効にすることができません。</p>

10 事前設定済みの N-VDS の場合は、次の詳細を提供します。

オプション	説明
N-VDS の外部 ID	このノードが属するトランスポート ゾーンの N-VDS 名と同じにする必要があります。
VTEP	仮想トンネル エンドポイントの名前。

トランスポート ノードとしてホストを追加すると、NSX Controller とホストの接続がアップ状態に変わります。

11 [トランスポート ノード] ページで接続ステータスを表示します。

12 または、CLI コマンドを使用して、接続ステータスを表示します。

- ◆ ESXi の場合には、**esxcli network ip connection list | grep 1234** コマンドを実行します。

```
# esxcli network ip connection list | grep 1234
tcp    0    0 192.168.210.53:20514 192.168.110.34:1234 [ESTABLISHED] 1000144459
newreno netcpa
```

- ◆ KVM の場合には、**netstat -anp --tcp | grep 1234** コマンドを入力します。

```
user@host:~$ netstat -anp --tcp | grep 1234
tcp    0    0 192.168.210.54:57794 192.168.110.34:1234 [ESTABLISHED] -
```

- 13 (オプション) GET <https://<nsx-mgr>/api/v1/transport-nodes/<node-id>> API 呼び出しを使用して、トランスポート ノードを確認します。

```
{
  "resource_type": "TransportNode",
  "description": "",
  "id": "95c8ce77-f895-43de-adc4-03a3ae2565e2",
  "display_name": "node-comp-01b",
  "tags": [],
  "transport_zone_endpoints": [
    {
      "transport_zone_id": "efd7f38f-c5da-437d-af03-ac598f82a9ec",
      "transport_zone_profile_ids": [
        {
          "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
          "resource_type": "BfdHealthMonitoringProfile"
        }
      ]
    }
  ],
  "host_switches": [
    {
      "host_switch_profile_ids": [
        {
          "value": "8abdb6c0-db83-4e69-8b99-6cd85bfcc61d",
          "key": "UplinkHostSwitchProfile"
        },
        {
          "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
          "key": "LldpHostSwitchProfile"
        }
      ],
      "host_switch_name": "overlay-hostswitch",
      "pnics": [
        {
          "device_name": "vmnic1",
          "uplink_name": "uplink-1"
        }
      ],
      "static_ip_pool_id": "c78ac522-2a50-43fe-816a-c459a210127e"
    }
  ],
  "node_id": "c551290a-f682-11e5-ae84-9f8726e1de65",
  "_create_time": 1460051753373,
  "_last_modified_user": "admin",
  "_system_owned": false,
  "_last_modified_time": 1460051753373,
  "_create_user": "admin",
  "_revision": 0
}
```

14 新しく作成したトランスポート ノードをトランスポート ゾーンに追加します。

- a トランスポート ノードを選択します。
- b [アクション]-[トランスポート ゾーンに追加] の順に選択します。
- c ドロップダウン メニューからトランスポート ゾーンを選択します。

他のフィールドの値はすべて設定されています。

注: 標準 N-VDS の場合、トランスポート ノードの作成後に、トンネル エンドポイントへの IP アドレスの割り当てなどの設定を変更するには、ホストの CLI ではなく NSX Manager の GUI から行う必要があります。

次のステップ

ネットワーク インターフェイスを vSphere 標準スイッチから NSX-T 仮想分散スイッチに移行します。[「[VMkernel の N-VDS スイッチへの移行](#)」] を参照してください。

トランスポート ノードの自動作成の設定

vCenter Server クラスタがある場合、単一または複数のクラスタのすべての NSX-T ホストで、トランスポート ノードのインストールと作成を自動化できます。

注: NSX-T トランスポート ノードの自動作成は、vCenter Server 6.5 Update 1、6.5 Update 2 および 6.7 でのみ使用できます。

トランスポート ノードがすでに設定されている場合は、そのノードではトランスポート ノードの自動作成は実行できません。

前提条件

- ホストは、vCenter Server クラスタを構成している必要があります。
- トランスポート ゾーンが設定されている必要があります。
- アップリンク プロファイルが設定されている必要があります。設定されていない場合は、デフォルトのアップリンク プロファイルを使用できます。
- IP アドレス プールが設定されているか、ネットワーク環境 DHCP が使用できる必要があります。
- ホスト上で 1 個以上の未使用の物理 NIC が必要です。
- vCenter Server には、1 つ以上のクラスタが必要です。
- コンピュート マネージャを設定する必要があります。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック]-[ノード]-[ホスト] の順に選択します。
- 3 ドロップダウン メニューの [管理対象] から既存のコンピュート マネージャを選択します。
- 4 クラスタを選択して、[クラスタの設定] をクリックします。

5 クラスタの詳細をすべて指定します。

オプション	説明
NSX を自動的にインストール	ボタンを切り替えて、vCenter Server クラスタのすべての NSX-T ホストでインストールを有効にします。
トランスポート ノードを自動的に作成	<p>ボタンを切り替えて、vCenter Server クラスタのすべてのホストでトランスポート ノードの作成を有効にします。この設定は必須です。</p> <p>注: 事前構成済みのトランスポート ノードがクラスタ内にある場合、または別のクラスタに移動されている場合、NSX-T は、クラスタのトランスポート ノードテンプレートで定義されている構成を使用して事前構成済みのトランスポート ノードを更新しません。すべてのノードで同じ構成が使用されるようにするには、事前構成済みのトランスポート ノードを削除して、ホストをクラスタに追加します。</p>
トランスポート ゾーン	ドロップダウン メニューから既存のトランスポート ノードを選択します。
アップリンク プロファイル	<p>ドロップダウン メニューから既存のアップリンク プロファイルを選択するか、アップリンクのカスタム プロファイルを作成します。</p> <p>注: クラスタ内のホストには同じアップリンク プロファイルが必要です。</p> <p>デフォルトのアップリンク プロファイルも使用できます。</p>
IP アドレスの割り当て	<p>ドロップダウン メニューから [DHCP を使用] または [IP アドレス プールを使用] のいずれかを選択します。</p> <p>[IP アドレス プールを使用] を選択する場合には、ドロップダウン メニューを使用して、ネットワーク内の既存の IP アドレス プールを割り当てる必要があります。</p>
物理 NIC	<p>物理 NIC が使用されていないことを確認します。標準の vSwitch または vSphere Distributed Switch などが使用していることがあります。すでに使用されている場合、トランスポート ノードは部分的成功状態になり、ファブリック ノードの LCP 接続の確立に失敗します。</p> <p>デフォルトのアップリンクを使用することも、ドロップダウン メニューから既存のアップリンクを割り当てることもできます。</p> <p>[物理 NIC の追加] をクリックして、環境内の NIC の数を増やします。</p>

クラスタ内の各ホストでは、NSX-T のインストールとトランスポート ノードの作成は並行して行われます。プロセス全体は、クラスタ内のホスト数によって異なります。

新しいホストが vCenter Server クラスタに追加されると、NSX-T のインストールとトランスポート ノードの作成が自動的に実行されます。

6 (オプション) ESXi の接続状況を確認します。

```
# esxcli network ip connection list | grep 1234
tcp    0    0 192.168.210.53:20514 192.168.110.34:1234 [ESTABLISHED] 1000144459
newreno netcpa
```

7 (オプション) クラスタ内のホストから NSX-T のインストールとトランスポート ノードを削除します。

- クラスタを選択して、[クラスタの設定] をクリックします。
- [NSX を自動的にインストール] ボタンを切り替えて、オプションを無効にします。
- 1 台以上のホストを選択し、[NSX のアンインストール] をクリックします。

アンインストールには最大で 3 分ほどかかります。

リンク集約による ESXi ホスト トランスポート ノードの設定

この手順では、リンク集約グループが設定されたアップリンク プロファイルを作成する方法、およびそのアップリンク プロファイルを使用するように ESXi ホスト トランスポート ノードを設定する方法について説明します。

前提条件

- アップリンク プロファイルの作成手順について理解していること。「[「アップリンク プロファイルの作成」](#)」を参照してください。
- ホスト トランスポート ノードの作成手順について理解していること。「[「ホスト トランスポート ノードの作成」](#)」を参照してください。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック]-[プロファイル]-[アップリンク プロファイル] の順に選択し、[追加] をクリックします。
- 3 名前を入力します。必要に応じて説明も入力します。
たとえば、「**uplink-profile1**」という名前を入力します。
- 4 [LAG] の [追加] をクリックして、リンク集約グループを追加します。
たとえば、2 つのアップリンクを持つ **lag1** という LAG を追加します。
- 5 [チーミング] の [デフォルトのチーミング] エントリに「[アクティブ アップリンク]」と入力します。
- 6 [チーミング] で [追加] をクリックして、リンク集約のエントリを追加します。
 - a 名前を入力します。
 - b チーミング ポリシーを選択します。
 - c [アクティブ アップリンク] フィールドに、手順 4 で追加した LAG の名前を入力します。この例では、LAG 名は **lag1** です。
 - d [トランスポート VLAN] および [MTU] の値を入力します。
 - e ダイアログ ボックスの下部にある [追加] をクリックします。
- 7 [ファブリック]-[ノード]-[トランスポート ノード]-[追加] の順に選択します。
- 8 [全般] タブに情報を入力します。
- 9 [N-VDS] タブで、手順 3 で作成したアップリンク プロファイル **uplink-profile1** を選択します。
- 10 [物理 NIC] フィールドに、物理 NIC のドロップダウン リスト、およびアップリンク プロファイルを作成したときに指定したアップリンクのドロップダウン リストが表示されます。ここでは、手順 4 で作成された LAG **lag1** に対応するアップリンク **lag1-0** および **lag1-1** が表示されます。**lag1-0** の物理 NIC および **lag1-1** の物理 NIC を選択します。
- 11 その他のフィールドの情報を入力します。

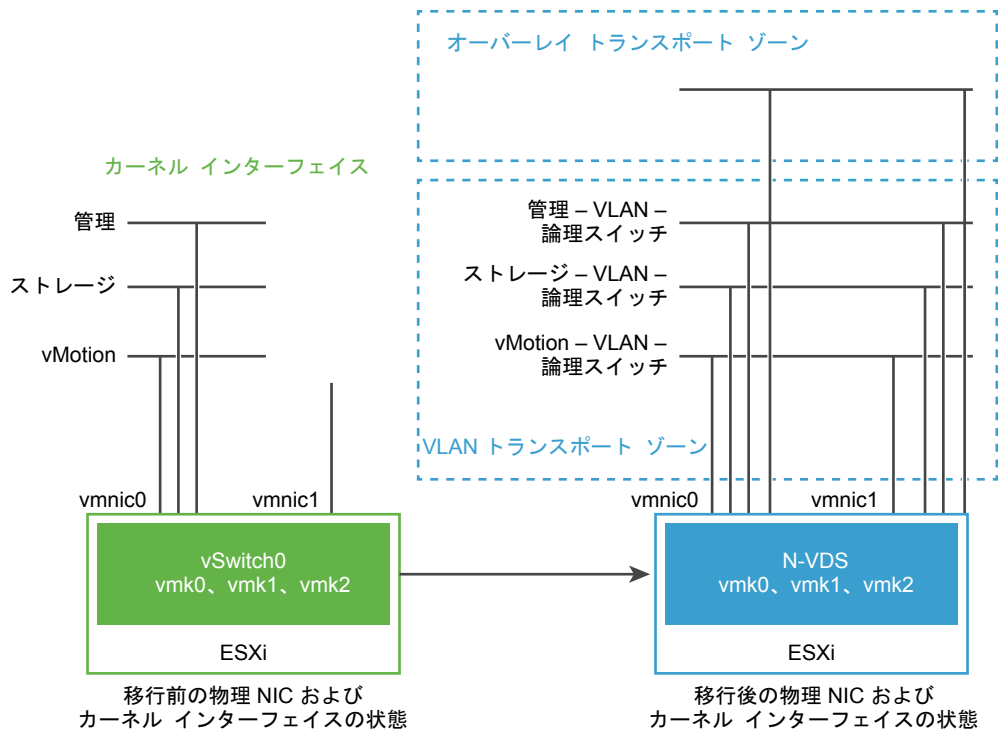
VMkernel の N-VDS スイッチへの移行

トランスポート ノードを作成するときに、物理 NIC およびカーネル インターフェイスの vSphere 標準スイッチ (VSS) から NSX-T 仮想分散スイッチ (N-VDS) への移行が必要になる場合があります。移行後、N-VDS は VLAN ネットワーク上のトラフィックを処理します。

物理 NIC とその VMkernel インターフェイスは、最初は vSphere ESXi ホストの VSS に接続します。これらのカーネル インターフェイスは、これらのホストで定義され、管理インターフェイス、ストレージ、その他のインターフェイスへの接続を提供します。移行後は、VMkernel インターフェイスと関連付けられた物理 NIC が N-VDS に接続され、VLAN トランスポート ゾーンとオーバーレイ トランスポート ゾーンのトラフィックを処理します。

次の図では、ホストの物理 NIC が 2 つのみの場合、冗長性を確保するために両方の NIC を N-VDS に割り当てることができます。

図 8-2. N-VDS へのネットワーク インターフェイスの移行前と移行後



移行前は、vSphere ESXi ホストに 2 つの物理ポート (vmnic0 と vmnic1) から派生した 2 つのアップリンクがあります。このとき、vmnic0 はアクティブな状態に設定され、VSS に接続されますが、vmnic1 は使用されません。さらに、vmk0、vmk1、vmk2 という 3 つの VMkernel インターフェイスがあります。

NSX-T Manager ユーザー インターフェイスまたは NSX-T API を使用して、VMkernel インターフェイスを移行します。NSX-T API ガイド を参照してください。

移行後には、vmnic0、vmnic1、およびそれらの VMkernel インターフェイスが N-VDS スイッチに移行されます。vmnic0 と vmnic1 の両方が、VLAN およびオーバーレイ トランスポート ゾーンを介して接続されます。

NSX-T Manager ユーザー インターフェイスを使用した VMkernel インターフェイスの N-VDS スイッチへの移行

NSX-T Manager ユーザー インターフェイスを使用すると、VSS から N-VDS スイッチに管理インターフェイスを含むすべてのカーネル インターフェイスを移行できます。

この例では、2 つの物理アダプタ vmnic0 と vmnic1 を持つ vSphere ESXi ホストについて考えてみます。ホスト上のデフォルトの VSS スイッチには、vmnic0 にマッピングされた単一アップリンクが設定されています。VSS には、ノード上で管理トラフィックを実行するための VMkernel インターフェイス vmk0 も設定されています。この例の目的は、vmnic0 および vmk0 を N-VDS スイッチに移行することです。

ホストの準備の一環として、管理トラフィックと仮想マシンのトラフィックをそれぞれ実行する VLAN トランスポート ゾーンおよびオーバーレイトランスポート ゾーンが作成されます。また、N-VDS スイッチも作成され、アップリンクが vmnic1 にマッピングされた状態で設定されます。移行後、NSX-T は vmnic0 と vmk0 の両方を VSS からノード上の N-VDS スイッチに移行します。

前提条件

- 物理ネットワーク インフラストラクチャによって提供される LAN 接続が、vmnic0 と vmnic1 の間で同じであることを確認します。
- 未使用の物理 NIC である vmnic1 に、vmnic0 とのレイヤー 2 接続があることを確認します。
- この移行に含まれるすべての VMkernel インターフェイスが同じネットワークに属していることを確認します。異なるネットワークに接続されているアップリンクに VMkernel を移行すると、ホストがアクセスできなくなったり、ホストが機能しなくなる可能性があります。

手順

- 1 NSX Manager ユーザー インターフェイスで、[ファブリック] -> [プロファイル] -> [アップリンク プロファイル] の順に移動します。
- 2 アクティブ アップリンクとして vmnic0 を、パッシブ アップリンクとして vmnic1 を使用して、アップリンク プロファイルを作成します。
- 3 [ファブリック] -> [トランスポート ゾーン] -> [追加] の順に移動します。
- 4 仮想マシン トラフィックおよび管理トラフィックをそれぞれ処理するためのオーバーレイ トランスポート ゾーンおよび VLAN トランスポート ゾーンを作成します。

注: VLAN トランスポート ゾーンとオーバーレイ トランスポート ゾーンで使用する N-VDS 名を一致させます。

- 5 [ファブリック] -> [ノード] -> [トランスポート ノード] の順に移動します。
- 6 トランスポート ノードに両方のトランスポート ゾーンを追加します。
- 7 N-VDS タブで、N-VDS で使用されるアップリンク、物理アダプタを定義して N-VDS を追加します。
トランスポート ノードは、単一のアップリンクを介してトランスポート ゾーンに接続されます。
- 8 移行後に vmk0 および vmnic0 が VLAN トランスポート ゾーンに接続されるようにするには、該当する VLAN トランスポート ゾーンの論理スイッチを作成します。

- 9 トランスポート ノードを選択して、[アクション] -> [ESX の VMkernel アダプタおよび物理アダプタの移行] の順にクリックします。
- 10 [論理スイッチに移行] を選択します。
- 11 N-VDS スイッチを選択します。
- 12 VMkernel アダプタおよび関連する論理スイッチを追加します。
- 13 VMkernel インターフェイスに対応する物理アダプタを追加します。1 つ以上の物理アダプタが VSS スイッチ上に残っていることを確認します。
- 14 [保存] をクリックします。
- 15 [続行] をクリックして移行を開始します。
- 16 NSX Manager から vmnic0 および vmk0 への接続をテストします。
- 17 または、vCenter Server で、VMkernel アダプタが NSX-T スイッチに関連付けられていることを確認します。

VMkernel インターフェイスおよび対応する物理アダプタが N-VDS に移行されます。

次のステップ

VSS または VDS スイッチに VMkernel の移行を戻すことができます。

NSX-T Manager ユーザー インターフェイスを使用して VMkernel インターフェイスの移行を VSS スイッチに戻す

VMkernel インターフェイスの移行を VSS スイッチに戻すには、ESXi ホスト上にポート グループがあることを確認します。

NSX-T で VMkernel インターフェイスを N-VDS スイッチから VSS スイッチに移行するには、ポート グループが必要です。ポート グループは、これらのインターフェイスを VSS スイッチに移行するよう求めるネットワーク要求を受け入れます。この移行に参加するポート メンバーは、バンド幅およびポリシー設定に基づいて決定されます。

VMkernel 移行を VSS スイッチに戻す前に、VMkernel インターフェイスが機能していて、N-VDS スイッチ上で接続が稼動中であることを確認します。

前提条件

- vSphere ESXi サーバ上にポート グループがあります。

手順

- 1 NSX Manager ユーザー インターフェイスで、[ファブリック] -> [ノード] -> [トランスポート ノード] の順に移動します。
- 2 トランスポート ノードを選択して、[アクション] -> [ESX の VMkernel アダプタおよび物理アダプタの移行] の順にクリックします。
- 3 [ポート グループに移行] を選択します。
- 4 N-VDS スイッチを選択します。
- 5 VMkernel アダプタおよび関連する論理スイッチを追加します。

- 6 VMkernel インターフェイスに対応する物理アダプタを追加します。1 つ以上の物理アダプタが VSS スイッチ上に残っていることを確認します。
- 7 [保存] をクリックします。
- 8 [続行] をクリックして移行を開始します。
- 9 NSX Manager から vmnic0 および vmk0 への接続をテストします。
- 10 または、vCenter Server で、VMkernel アダプタが VSS スイッチに関連付けられていることを確認します。

VMkernel インターフェイスおよび対応する物理アダプタが N-VDS に移行されます。

次のステップ

API を使用して VMkernel インターフェイスを移行することができます。「[「API を使用した N-VDS へのカーネル インターフェイスの移行」](#)」を参照してください。

API を使用した N-VDS へのカーネル インターフェイスの移行

NSX-T API を使用する場合は、必ずすべてのカーネル インターフェイスを移行してから、管理インターフェイスを移行してください。

ホストの 2 つのアップリンクがそれぞれの物理 NIC に接続されているとします。この手順ではまず、ストレージのカーネル インターフェイス vmk1 を N-VDS へ移行するところから開始します。このカーネル インターフェイスが N-VDS に正常に移行された後で、管理カーネル インターフェイスを移行できます。

NSX-T API ガイド を参照してください。

前提条件

- 物理ネットワーク インフラストラクチャによって提供される LAN 接続が、vmnic0 と vmnic1 の間で同じであることを確認します。
- 未使用の物理 NIC である vmnic1 に、vmnic0 とのレイヤー 2 接続があることを確認します。
- この移行に含まれるすべての VMkernel インターフェイスが同じネットワークに属していることを確認します。異なるネットワークに接続されているアップリンクに VMkernel を移行すると、ホストがアクセスできなくなったり、ホストが機能しなくなる可能性があります。

手順

- 1 OVERLAY トランスポート ゾーンで使用される N-VDS の host_switch_name を持つように、VLAN トランスポート ゾーンを作成します。
- 2 VSS 上の vmk1 で使用される VLAN ID に一致する VLAN ID を持つ VLAN トランスポート ゾーンに、VLAN にバックアップされる論理スイッチを作成します。
- 3 VLAN トランスポート ゾーンに vSphere ESXi トランスポート ノードを追加します。
- 4 vSphere ESXi トランスポート ノードの設定を取得します。

GET /api/v1/transport-nodes/<transportnode-id>

<<transportnode-id>> はトランスポート ノードの UUID です。

- 5 vmk1 を N-VDS に移行します。

```
PUT https://<NSXmgr>/api/v1/transport-nodes/<transportnode-id> ?
if_id=<vmk>&esx_mgmt_if_migration_dest=<network>
```

<<transportnode-id>> はトランスポート ノードの UUID です。<<vmk>> は VMkernel インターフェイス vmk1 の名前です。<<network>> は移行先の論理スイッチの UUID です。

- 6 移行が正常に終了したことを確認します。

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

移行の状態が **SUCCESS** と表示されるまで待ちます。vCenter Server で VMkernel インターフェイスの移行の状態を確認することもできます。

VMkernel インターフェイスが VSS から N-VDS スイッチに移行されました。

次のステップ

残りの VMkernel インターフェイスおよび VSS の管理カーネル インターフェイスを N-VDS に移行できます。

API を使用した VSS から N-VDS への管理カーネル インターフェイスの移行

他のすべてのカーネル インターフェイスを移行したら、管理カーネル インターフェイスの移行に進みます。管理カーネル インターフェイスを移行する際は、vmnic0 と vmk0 を VSS から N-VDS に移動します。

これにより、物理アップリンク vmnic0 と vmk0 を 1 つのステップでまとめて N-VDS に移行できます。vmnic0 がそのアップリンクの 1 つとして構成されるように、トランスポート ノードの設定を変更します。

注: アップリンク vmnic0 とカーネル インターフェイス vmk0 を別々に移行する場合は、まず vmk0 を移行し、その後 vmnic0 を移行します。最初に vmnic0 を移行すると、vmk0 はバックアップアップリンクがない状態で VSS に残り、ホストへの接続が失われます。

前提条件

- 移行済みの vmknics への接続を確認します。[「API を使用した N-VDS へのカーネル インターフェイスの移行」](#)を参照してください。
- vmk0 と vmk1 で別の VLAN を使用している場合は、PNIC vmnic0 と vmnic1 に接続されている物理スイッチのトランク VLAN を、両方の VLAN をサポートするように設定する必要があります。
- ストレージ VLAN によってバックアップされる論理スイッチの vmk1 インターフェイスと、vMotion VLAN によってバックアップされる論理スイッチの vmk2 インターフェイスに、外部デバイスがアクセスできることを確認します。

手順

- 1 (オプション) VSS に 2 つ目の管理カーネル インターフェイスを作成し、この新しく作成されたインターフェイスを N-VDS に移行します。
- 2 (オプション) 外部デバイスで、テスト管理インターフェイスへの接続を確認します。

- 3 vmk0（管理インターフェイス）と vmk1（ストレージ インターフェイス）で別の VLAN を使用している場合は、VSS 上の vmk0 で使用される VLAN ID に一致する VLAN ID を持つ VLAN トランスポート ゾーンに、VLAN にバッキングされる論理スイッチを作成します。
- 4 vSphere ESXi トランスポート ノードの設定を取得します。

GET /api/v1/transport-nodes/<transportnode-id>

<<transportnode-id>> はトランスポート ノードの UUID です。

- 5 設定の **host_switch_spec:host_switches** 要素で、pnics テーブルに vmnic0 を追加し、それを専用のアップリンク、uplink-2 に割り当てます。

注: VMkernel インターフェイスを移行中に、vmnic1 を uplink-1 に割り当てました。移行を成功させ、移行後にホストにアクセスできるようにするためには、vmnic0（管理インターフェイス）を専用のアップリンクに割り当てする必要があります。

```
"pnics": [
  {
    "device_name": "vmnic0",
    "uplink_name": "uplink-2"
  },
  {
    "device_name": "vmnic1",
    "uplink_name": "uplink-1"
  }
],
```

- 6 更新された設定を使用して、管理カーネル インターフェイス vmk0 を N-VDS に移行します。

PUT api/v1/transport-nodes/< transportnode-id>?if_id=<vmk>&esx_mgmt_if_migration_dest=<network>

<<transportnode-id>> はトランスポート ノードの UUID です。<<vmk>> は VMkernel 管理インターフェイス vmk0 の名前です。<Network> は移行先の論理スイッチの UUID です。

- 7 移行が正常に終了したことを確認します。

GET /api/v1/transport-nodes/<transportnode-id>/state

移行の状態が **SUCCESS** と表示されるまで待ちます。vCenter Server で、カーネル アダプタが新しい論理スイッチ名を表示するように設定されていることを確認できます。

次のステップ

カーネル インターフェイスおよび管理インターフェイスの移行を N-VDS から VSS に戻すよう選択することもできます。

API を使用して N-VDS スイッチから VSS スイッチ VMkernel インターフェイスの移行を戻す

VMkernel インターフェイスを元に戻す場合は、まず管理カーネル インターフェイスを移行する必要があります。その後で、他のカーネル インターフェイスを N-VDS から VSS スイッチに移行します。

手順

- 1 トランスポート ノードの状態が正常であることを確認します。

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

- 2 vSphere ESXi トランスポート ノードの設定を取得し、"host_switch_spec":"host_switches" 要素内に定義されている物理 NIC を検索します。

```
GET /api/v1/transport-nodes/<transportnode-id>
```

```
"pnics": [
  { "device_name": "vmnic0",
    "uplink_name": "uplink-2"
  },
  { "device_name": "vmnic1",
    "uplink_name": "uplink-1"
  }
],
```

- 3 トランスポート ノードの設定にある "host_switch_spec":"host_switches" 要素から vmnic0 を削除し、管理インターフェイスの移行を準備します。

```
"pnics": [
  { "device_name": "vmnic1",
    "uplink_name": "uplink-1"
  }
],
```

- 4 変更した設定を使用して、管理インターフェイス vmnic0 と vmk0 を N-VDS から VSS に移行します。

```
PUT api/v1/transport-nodes/< transportnode-id>?
if_id=vmk0&esx_mgmt_if_migration_dest=<vmk0_port_group_name>
```

ここで、<<vmk0_port_group>> は、論理スイッチに移行する前に vmk0 に割り当てられたポート グループの名前です。

- 5 移行の状態を確認します。

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

状態が「SUCCESS」と表示されるまで待ちます。

- 6 vSphere ESXi トランスポート ノードの設定を取得します。

```
GET /api/v1/transport-nodes/<transportnode-id>
```

- 7 前述のトランスポート ノードの設定を使用して、vmk1 を N-VDS から VSS に移行します。

```
PUT api/v1/transport-nodes/< transportnode-id>?
if_id=vmk1&esx_mgmt_if_migration_dest=<vmk1_port_group>
```

ここで、<<vmk1_port_group>> は、論理スイッチに移行する前に vmk1 に割り当てられたポート グループの名前です。

注: VSS には関連付けられた物理 NIC がないため、物理 NIC が 1 つ以上搭載された vmk0 または vmk1 を VSS に移行する必要があります。

- 8 トランスポート ノードの状態が正常であることを確認します。

GET /api/v1/transport-nodes/<transportnode-id>/state.

- 9 移行後の確認を実行して、問題が発生していれば回避します。
 - a VSS にアップリンク インターフェイスを接続するまでは、管理カーネル インターフェイス vmk0 は移行しないでください。
 - b 確実に vmk0 が vmnic0 から IP アドレスを受信するようにします。受信しない場合は、IP アドレスが変更され、古い IP アドレスを使用しているために vCenter Server のような他のコンポーネントとホスト間の接続が失われている可能性があります。

トランスポート ノードの状態の確認

トランスポート ノードの作成プロセスが正常に機能していることを確認します。

ホスト トランスポート ノードの作成後、ホスト上に N-VDS を配置します。

手順

- 1 NSX-T にログインします。
- 2 [トランスポート ノード] ページに移動し、N-VDS の状態を確認します。
- 3 または、**esxcli network ip interface list** コマンドを使用して、ESXi 上の N-VDS を確認します。

ESXi でのコマンドの出力には、Distributed Switch (VDS) の名前がついた vmk インターフェイス (vmk10 など) が含まれます。この Distributed Switch の名前は、トランスポート ゾーンとトランスポート ノードを設定する際に使用した名前です。

```
# esxcli network ip interface list
...

vmk10
  Name: vmk10
  MAC Address: 00:50:56:64:63:4c
  Enabled: true
  Portset: DvsPortset-1
  Portgroup: N/A
  Netstack Instance: vxlan
  VDS Name: [overlay-hostswitch]
  VDS UUID: 18 ae 54 04 2c 6f 46 21-b8 ae ef ff 01 0c aa c2
  VDS Port: 10
  VDS Connection: 10
  Opaque Network ID: N/A
  Opaque Network Type: N/A
  External ID: N/A
```

```
MTU: 1600
TSO MSS: 65535
Port ID: 67108895
```

```
...
```

vSphere Client を使用している場合、ユーザー インターフェイスでホストの [設定] > [ネットワーク アダプタ] を順に選択し、インストールされている N-VDS を確認できます。

N-VDS を確認するための KVM のコマンドは、**ovs-vsctl show** です。KVM では、N-VDS の名前は nsx-switch.0 と表示されます。トランスポート ノードの設定で使った名前とは異なる点に注意してください。これは仕様です。

```
# ovs-vsctl show
...
    Bridge "nsx-switch.0"
        Port "nsx-uplink.0"
            Interface "em2"
        Port "nsx-vtep0.0"
            tag: 0
            Interface "nsx-vtep0.0"
                type: internal
        Port "nsx-switch.0"
            Interface "nsx-switch.0"
                type: internal
    ovs_version: "2.4.1.3340774"
```

4 トランスポート ノードに割り当てられているトンネル エンドポイント アドレスを確認します。

次に示すように、vmk10 のインターフェイスは、NSX-T IP アドレス プールまたは DHCP から IP アドレスを受け取ります。

```
# esxcli network ip interface ipv4 get
```

Name	IPv4 Address	IPv4 Netmask	IPv4 Broadcast	Address Type	DHCP	DNS
vmk0	192.168.210.53	255.255.255.0	192.168.210.255	STATIC		false
vmk1	10.20.20.53	255.255.255.0	10.20.20.255	STATIC		false
[vmk10	192.168.250.3]	255.255.255.0	192.168.250.255	STATIC		false

KVM では、**ifconfig** コマンドを使用して、トンネル エンドポイントと IP アドレス割り当てを確認できます。

```
# ifconfig
...
nsx-vtep0.0 Link encap:Ethernet HWaddr ba:30:ae:aa:26:53
    inet addr:[192.168.250.4] Bcast:192.168.250.255 Mask:255.255.255.0
...
```

5 API で状態を確認します。

GET <https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>/state> API 呼び出しを使用します。次はその例です。

```
{
  "state": "success",
  "host_switch_states": [
    {
      "endpoints": [
        {
          "default_gateway": "192.168.250.1",
          "device_name": "vmk10",
          "ip": "192.168.250.104",
          "subnet_mask": "255.255.255.0",
          "label": 69633
        }
      ],
      "transport_zone_ids": [
        "efd7f38f-c5da-437d-af03-ac598f82a9ec"
      ],
      "host_switch_name": "overlay-hostswitch",
      "host_switch_id": "18 ae 54 04 2c 6f 46 21-b8 ae ef ff 01 0c aa c2"
    }
  ],
  "transport_node_id": "2d030569-5769-4a13-8918-0c309c63fdb9"
}
```

コンピュート マネージャの追加

コンピュート マネージャは、vCenter Server のように、ホストや仮想マシンなどのリソースを管理するアプリケーションです。NSX-T は、コンピュート マネージャをポーリングし、ホストまたは仮想マシンの追加や削除などの変更を検出し、インベントリを更新します。NSX-T は、スタンドアローンのホストや仮想マシンなど、コンピュート マネージャがなくてもインベントリ情報を取得するため、コンピュート マネージャの追加はオプションです。

今回のリリースでは、この機能は次のものをサポートしています。

- vCenter Server バージョン 6.5 Update 1、6.5 Update 2、および 6.7。
- vCenter Server との IPv6 または IPv4 による通信。
- 最大 5 個のコンピュート マネージャ。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 ナビゲーション パネルから、[ファブリック] - [コンピュート マネージャ] の順に選択します。
- 3 [追加] をクリックします。

4 コンピュート マネージャの詳細を設定します。

オプション	説明
名前と説明	vCenter Server を識別する名前を入力します。 必要に応じて、vCenter Server のクラスタ数などの詳細を入力します。
ドメイン名/IP アドレス	vCenter Server の IP アドレスを入力します。
タイプ	デフォルトのオプションを使用します。
ユーザー名とパスワード	vCenter Server ログイン認証情報を入力します。
サムプリント	vCenter Server SHA-256 サムプリント アルゴリズムの値を入力します。

サムプリント値を空白にすると、サーバのサムプリントを使用するように指示されます。

サムプリントを受け入れてから NSX-T が vCenter Server リソースを検出して登録するまで、数秒かかります。

5 進行状況アイコンが [処理中] から [未登録] に変わった場合は、次の手順を実行してエラーを解決します。

- a エラー メッセージを選択し、[解決] をクリックします。次のようなエラー メッセージが表示される可能性があります：

Extension already registered at CM <vCenter Server name> with id <extension ID>

- b vCenter Server 認証情報を入力し、[解決] をクリックします。

すでに登録がされている場合には置き換えられます。

[コンピュー ト マネージャ] パネルに、コンピュー ト マネージャのリストが表示されます。マネージャの名前をクリックすると、マネージャの詳細を表示して編集できます。また、マネージャに適用するタグを管理できます。

Network I/O Control の設定

Network I/O Control (NIOC) プロファイルを使用して、ビジネス上不可欠なアプリケーションにネットワーク バンド幅を割り当てたり、いくつかの種類のトラフィックが共通のリソースで競合する問題を解決したりします。

NIOC プロファイルは、ホスト上の物理アダプタのキャパシティに基づいて、システム トラフィックのバンド幅を予約するメカニズムを導入しています。Network I/O Control のバージョン 3 の機能では、ネットワーク リソース予約とスイッチ全体への割り当てが向上しています。

NSX-T 向け Network I/O Control バージョン 3 は、仮想マシンや、vSphere Fault Tolerance などのインフラストラクチャ サービスに関連するシステム トラフィックのリソース管理をサポートします。システム トラフィックは、vSphere ESXi ホストに完全に関連付けられています。

システム トラフィックに対するバンド幅の確保

Network I/O Control バージョン 3 では、シェア、予約、および制限の構造を使用して、仮想マシンのネットワーク アダプタにバンド幅をプロビジョニングします。これらの構造は、NSX-T Manager ユーザー インターフェイスで定義できます。仮想マシン トラフィックのバンド幅予約は、アドミッション コントロールでも使用されます。仮想マシンをパワーオンすると、アドミッション コントロール ユーティリティは、十分なバンド幅が使用できることを確認してから、リソース キャパシティの提供が可能なホストに仮想マシンを配置します。

システム トラフィックのバンド幅割り当て

vSphere Fault Tolerance、vSphere vMotion、仮想マシンなどによって生成されるトラフィックに一定量のバンド幅を割り当てるように Network I/O Control を構成できます。

- 管理トラフィック：ホスト管理のトラフィックです。
- Fault Tolerance (FT) トラフィック：フェイルオーバーとリカバリのトラフィックです。
- NFS トラフィック：ネットワーク ファイル システムでのファイル転送に関連したトラフィックです。
- vSAN トラフィック：仮想ストレージ エリア ネットワークによって生成されるトラフィックです。
- vMotion トラフィック：コンピューティング リソースの移行トラフィックです。
- vSphere Replication トラフィック：レプリケーションのトラフィックです。
- vSphere Data Protection バックアップ トラフィック：データのバックアップによって生成されるトラフィックです。
- 仮想マシン トラフィック：仮想マシンによって生成されるトラフィックです。
- iSCSI トラフィック：iSCSI (Internet Small Computer System Interface) のトラフィック。

vCenter Server は、Distributed Switch の割り当てを、スイッチに接続されているホストの各物理アダプタに伝達します。

システム トラフィックのバンド幅割り当てパラメータ

Network I/O Control サービスでは、いくつかの構成パラメータを使用して、vSphere システムの基本機能からのトラフィックにバンド幅を割り当てます。システム トラフィックの割り当てパラメータ。

システム トラフィックの割り当てパラメータ

- シェア：シェアは、同じ物理アダプタ上で有効な他のシステム トラフィック タイプを基に、システム トラフィック タイプの相対的な優先度を 1 から 100 で示します。システム トラフィック タイプに割り当てられた相対的なシェアと、他のシステム機能で転送されたデータの量により、システム トラフィック タイプに使用できるバンド幅が決まります。
- 予約：単一の物理アダプタ上で確保する必要のある最小バンド幅 (Mbps)。すべてのシステム トラフィック タイプで予約される合計バンド幅は、最低キャパシティを備えた物理ネットワーク アダプタが提供できるバンド幅の 75% を超過することはできません。未使用の予約バンド幅は、システム トラフィックの他のタイプで利用できるようになります。ただし、Network I/O Control では、システム トラフィックが使用しないキャパシティを仮想マシンの配置に再配分しません。
- 制限：単一物理アダプタでシステム トラフィック タイプが使用できる最大バンド幅 (Mbps)。

注： 物理ネットワーク アダプタのバンド幅は最大 75% まで予約することができます。たとえば、10 GbE ネットワーク アダプタが ESXi ホストに接続されている場合、各トラフィック タイプには 7.5 Gbps のバンド幅のみを割り当てることができます。未予約の容量が多く残ることがあります。ホストは、未予約のバンド幅をシェア、制限、使用量に応じて動的に割り当てることができます。ホストは、システム機能の処理に十分なバンド幅のみを予約します。

N-VDS スイッチのシステム トラフィックに対する Network I/O Control および バンド幅割り当ての設定

NSX-T のホストで実行されるシステム トラフィックに最小バンド幅を確保するには、NSX-T の Distributed Switch でネットワーク リソース管理を有効にして設定します。

手順

- 1 NSX Manager Manager にログインします。 <https://<nsx-manager-IP-address>>
- 2 [ファブリック] > [プロファイル] の順に移動します。
- 3 [Network I/O Control (NIOC) プロファイル] を選択します。
- 4 [+ 追加] をクリックします。
- 5 [新しい NIOC プロファイル] 画面で、必要な詳細を入力します。
 - a NIOC プロファイルの名前を入力します。
 - b [ステータス] を [有効] にします。
 - c [ホスト インフラストラクチャのトラフィック リソース] セクションで、トラフィック タイプを選択し、制限、シェア、予約の値を入力します。
- 6 [追加] をクリックします。

NIOC プロファイルのリストに新しい NIOC プロファイルが追加されます。

API を使用した Network I/O Control と N-VDS スイッチ上のシステム トラフィックのバンド幅割り当ての設定

NSX-T API を使用して、ネットワークと、ホスト上で実行しているアプリケーションのバンド幅を設定できます。

手順

- 1 ホストに対して、システム定義およびユーザー定義の両方のホスト スイッチ プロファイルを表示するよう問い合わせます。
- 2 GET https://<nsx-mgr>/api/v1/host-switch-profiles?include_system_owned=true。

次のサンプル応答では、ホストに適用されている NIOC プロファイルが表示されています。

```
{
  "description": "This profile is created for Network I/O Control (NIOC).",
  "extends": {
    "$ref": "BaseHostSwitchProfile"+
  },
  "id": "NiocProfile",
  "module_id": "NiocProfile",
  "polymorphic-type-descriptor": {
    "type-identifier": "NiocProfile"
  },
}
```

```

"properties": {
  "_create_time": {
    "$ref": "EpochMsTimestamp"+,
    "can_sort": true,
    "description": "Timestamp of resource creation",
    "readonly": true
  },
  "_create_user": {
    "description": "ID of the user who created this resource",
    "readonly": true,
    "type": "string"
  },
  "_last_modified_time": {
    "$ref": "EpochMsTimestamp"+,
    "can_sort": true,
    "description": "Timestamp of last modification",
    "readonly": true
  },
  "_last_modified_user": {
    "description": "ID of the user who last modified this resource",
    "readonly": true,
    "type": "string"
  },
  "_links": {
    "description": "The server will populate this field when returning the resource. Ignored
on PUT and POST.",
    "items": {
      "$ref": "ResourceLink"+
    },
    "readonly": true,
    "title": "References related to this resource",
    "type": "array"
  },
  "_protection": {
    "description": "Protection status is one of the following:
      PROTECTED – the client who retrieved the entity is not allowed to modify it.
      NOT_PROTECTED – the client who retrieved the entity is allowed to modify it
      REQUIRE_OVERRIDE – the client who retrieved the entity is a super user and can modify
it,
      but only when providing the request header X-Allow-Overwrite=true.
      UNKNOWN – the _protection field could not be determined for this entity.",
    "readonly": true,
    "title": "Indicates protection status of this resource",
    "type": "string"
  },
  "_revision": {
    "description": "The _revision property describes the current revision of the resource.
      To prevent clients from overwriting each other's changes, PUT operations must include
the
      current _revision of the resource,
      which clients should obtain by issuing a GET operation.

```

If the `_revision` provided in a PUT request is missing or stale, the operation will be rejected.",

```
"readonly": true,
"title": "Generation of this resource config",
"type": "int"
},
```

```
"_schema": {
  "readonly": true,
  "title": "Schema for this resource",
  "type": "string"
},
```

```
"_self": {
  "$ref": "SelfResourceLink"+,
  "readonly": true,
  "title": "Link to this resource"
},
```

```
"_system_owned": {
  "description": "Indicates system owned resource",
  "readonly": true,
  "type": "boolean"
},
```

```
"description": {
  "can_sort": true,
  "maxLength": 1024,
  "title": "Description of this resource",
  "type": "string"
},
```

```
"display_name": {
  "can_sort": true,
  "description": "Defaults to ID if not set",
  "maxLength": 255,
  "title": "Identifier to use when displaying entity in logs or GUI",
  "type": "string"
},
```

```
"enabled": {
  "default": true,
  "description": "The enabled property specifies the status of NIOC feature.
```

When enabled is set to true, NIOC feature is turned on and the bandwidth allocations specified for the traffic resources are enforced.

When enabled is set to false, NIOC feature is turned off and no bandwidth allocation is guaranteed.

By default, enabled will be set to true.",

```
"nsx_feature": "Nioc",
"required": false,
"title": "Enabled status of NIOC feature",
"type": "boolean"
```

```

    },

    "host_infra_traffic_res": {
      "description": "host_infra_traffic_res specifies bandwidth allocation for various
traffic resources.",
      "items": {
        "$ref": "ResourceAllocation"+
      },
      "nsx_feature": "Nioc",
      "required": false,
      "title": "Resource allocation associated with NiocProfile",
      "type": "array"
    },

    "id": {
      "can_sort": true,
      "readonly": true,
      "title": "Unique identifier of this resource",
      "type": "string"
    },

    "required_capabilities": {
      "help_summary":
        "List of capabilities required on the fabric node if this
profile is used.
        The required capabilities is determined by whether specific features are enabled
in the profile.",
      "items": {
        "type": "string"
      },
      "readonly": true,
      "required": false,
      "type": "array"
    },

    "resource_type": {
      "$ref": "HostSwitchProfileType"+,
      "required": true
    },

    "tags": {
      "items": {
        "$ref": "Tag"+
      },
    },

    "maxItems": 30,
    "title": "Opaque identifiers meaningful to the API user",
    "type": "array"
  }
},
"title": "Profile for Nioc",
"type": "object"
}

```

- 3 NIOC プロファイルがない場合は、新しい NIOC プロファイルを作成します。

POST <https://<nsx-mgr>/api/v1/host-switch-profiles>

```
{
  "description": "Specify limit, shares and reservation for all kinds of traffic.
  Values for limit and reservation are expressed in percentage. And for shares,
  the value is expressed as a number between 1-100.\n\nThe overall reservation among all
  traffic types should not exceed 75%.
  Otherwise, the API request will be rejected.",
  "id": "ResourceAllocation",
  "module_id": "NiocProfile",
  "nsx_feature": "Nioc",
  "properties": {
    "limit": {
      "default": -1.0,
      "description": "The limit property specifies the maximum bandwidth allocation for
      a given
      traffic type and is expressed in percentage. The default value for this
      field is set to -1 which means the traffic is unbounded for the traffic
      type. All other negative values for this property is not supported\nand will be
      rejected by the API.",
      "maximum": 100,
      "minimum": -1,
      "required": true,
      "title": "Maximum bandwidth percentage",
      "type": "number"
    },
    "reservation": {
      "default": 0.0,
      "maximum": 75,
      "minimum": 0,
      "required": true,
      "title": "Minimum guaranteed bandwidth percentage",
      "type": "number"
    },
    "shares": {
      "default": 50,
      "maximum": 100,
      "minimum": 1,
      "required": true,
      "title": "Shares",
      "type": "int"
    },
    "traffic_type": {
      "$ref": "HostInfraTrafficType+",
      "required": true,
      "title": "Resource allocation traffic type"
    }
  }
}
```

```

},

"title": "Resource allocation information for a host infrastructure traffic type",
"type": "object"

```

- 4 新規に作成された NIOC プロファイルの NIOC プロファイル ID を使用して、トランスポート ノードの設定を更新します。

PUT <https://<nsx-mgr>/api/v1/transport-nodes/<TN-id>>

```

{
  "resource_type": "TransportNode",
  "description": "Updated NSX configured Test Transport Node",
  "id": "77816de2-39c3-436c-b891-54d31f580961",
  "display_name": "NSX Configured TN",
  "host_switch_spec": {
    "resource_type": "StandardHostSwitchSpec",
    "host_switches": [
      {
        "host_switch_profile_ids": [
          {
            "value": "e331116d-f59e-4004-8cfd-c577ae563a",
            "key": "UplinkHostSwitchProfile"
          },
          {
            "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
            "key": "LldpHostSwitchProfile"
          },
          {
            "value": "b0185099-8003-4678-b86f-edd47ca2c9ad",
            "key": "NiocProfile"
          }
        ],
        "host_switch_name": "nsxvswitch",
        "pnics": [
          {
            "device_name": "vmnic1",
            "uplink_name": "uplink1"
          }
        ],
        "ip_assignment_spec": {
          "resource_type": "StaticIpPoolSpec",
          "ip_pool_id": "ecddcdde-4dc5-4026-ad4f-8857995d4c92"
        }
      }
    ],
    "transport_zone_endpoints": [
      {
        "transport_zone_id": "e14c6b8a-9edd-489f-b624-f9ef12afbd8f",
        "transport_zone_profile_ids": [
          {
            "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
            "resource_type": "BfdHealthMonitoringProfile"
          }
        ]
      }
    ]
  }
}

```

```

    }
  ]
}
],

"host_switches": [
{
  "host_switch_profile_ids": [
    {
      "value": "e331116d-f59e-4004-8cfd-c577aefe563a",
      "key": "UplinkHostSwitchProfile"
    },
    {
      "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
      "key": "LldpHostSwitchProfile"
    }
  ],

  "host_switch_name": "nsxvswitch",
  "pnics": [
    {
      "device_name": "vmnic1",
      "uplink_name": "uplink1"
    }
  ],
  "static_ip_pool_id": "ecddcdde-4dc5-4026-ad4f-8857995d4c92"
}
],
"node_id": "41a4eebd-d6b9-11e6-b722-875041b9955d",
"_revision": 0
}

```

- 5 `com.vmware.common.respools.cfg` セクションの NIOC プロファイル パラメータが更新されていることを確認します。

```
# [root@ host:] net-dvs -l
```

```

      switch 1d 73 f5 58 99 7a 46 6a-9c cc d0 93 17 bb 2a 48 (vswitch)
max ports: 2560
global properties:

com.vmware.common.opaqueDvs = true ,      propType = CONFIG
com.vmware.nsx.kcp.enable = true ,      propType = CONFIG
com.vmware.common.alias = nsxvswitch ,      propType = CONFIG
com.vmware.common.uplinkPorts: uplink1  propType = CONFIG
com.vmware.common.portset.mtu = 1600, propType = CONFIG
com.vmware.etherswitch.cdp = LLDP, listen  propType = CONFIG
com.vmware.common.respools.version = version3,  propType = CONFIG
com.vmware.common.respools.cfg:
netsched.pools.persist.ft:0:50:-1:255
netsched.pools.persist.hbr:0:50:-1:255
netsched.pools.persist.vmotion:0:50:-1:255
netsched.pools.persist.vm:0:100:-1:255
netsched.pools.persist.iscsi:0:50:-1:255

```

```

netsched.pools.persist.nfs:0:50:-1:255
netsched.pools.persist.mgmt:0:50:-1:255
netsched.pools.persist.vdp:0:50:-1:255
netsched.pools.persist.vsan:0:50:-1:255
propType = CONFIG

```

- 6 ホスト カーネルの NIOC プロファイルを確認します。

```
# [root@ host:] /get /net/portsets/DvsPortset-1/ports/50335755/niocVnicInfo
```

```

Vnic NIOC Info
{
    Uplink reserved on:vmnic4
    Reservation in Mbps:200
    Shares:50
    Limit in Mbps:4294967295
    World ID:1001400726
    vNIC Index:0
    Respool Tag:0
    NIOC Version:3
    Active Uplink Bit Map:15
    Parent Respool ID:netsched.pools.persist.vm
}

```

- 7 # [root@ host:] /get /net/portsets/DvsPortset-1/uplinks/vmnic4/niocInfo

```

Uplink NIOC Info
{
    Uplink device:vmnic4
    Link Capacity in Mbps:750
    vm respool reservation:275
    link status:1
    NetSched Ready:1
    Infrastructure reservation:0
    Total VM reservation:200
    Total vnics on this uplink:1
    NIOC Version:3
    Uplink index in BitMap:0
}

```

NIOC プロファイルは、NSX-T ホスト上で実行しているアプリケーションに事前定義されたバンド幅割り当てを使用
して設定されます。

NSX Edge トランスポート ノードの作成

トランスポート ノードは、NSX-T オーバーレイまたは NSX-T VLAN ネットワークに参加できるノードです。N-VDS
が含まれているノードは、トランスポート ノードとして機能します。そのようなノードとして NSX Edge がありま
すが、これに限定されるものではありません。この手順で、NSX Edge をトランスポート ノードとして追加します。

NSX Edge は、1 つのオーバーレイ トランスポート ゾーンおよび複数の VLAN トランスポート ゾーンに属することができます。仮想マシンから外部へのアクセスが必要な場合は、NSX Edge が、仮想マシンの論理スイッチが属しているのと同じトランスポート ゾーンに属している必要があります。通常、NSX Edge は 1 つ以上の VLAN トランスポート ゾーンに属して、アップリンク アクセスを提供します。

注: テンプレート仮想マシンを使用してトランスポート ノードを作成する場合は、ホストの `/etc/vmware/nsx/` に証明書がないことを確認してください。証明書がすでに存在する場合、netcpa エージェントは新しい証明書を作成しません。

前提条件

- NSX Edge が管理プレーンに追加され、[ファブリック] > [Edge] ページで MPA 接続が確立されている必要があります。「[\[NSX Edge の管理プレーンへの追加\]](#)」を参照してください。
- トランスポート ゾーンが設定されている必要があります。
- アップリンク プロファイルが設定されている必要があります。設定されていない場合は、ベア メタル NSX Edge ノード用のデフォルトのアップリンク プロファイルを使用できます。
- IP アドレス プールが設定されているか、ネットワーク環境内の IP アドレス プールを使用できる必要があります。
- ホストまたは NSX Edge ノード上で 1 個以上の未使用の物理 NIC が必要です。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック] - [ノード] - [トランスポート ノード] - [追加] の順に選択します。
- 3 NSX Edge トランスポート ノードの名前を入力します。
- 4 ドロップダウン リストから NSX Edge ファブリック ノードを選択します。
- 5 このトランスポート ノードが属するトランスポート ゾーンを選択します。

NSX Edge トランスポート ノードは 2 つ以上のトランスポート ゾーン (NSX-T 接続用のオーバーレイとアップリンク接続用の VLAN) に属します。

- 6 [N-VDS] タブをクリックし、N-VDS 情報を設定します。

オプション	説明
N-VDS 名	トランスポート ゾーンの作成時に設定した名前と一致する必要があります。
アップリンク プロファイル	ドロップダウン メニューからアップリンク プロファイルを選択します。 使用可能なアップリンクは、選択したアップリンク プロファイルでの設定によって異なります。
IP アドレスの割り当て	オーバーレイ N-VDS に [IP アドレス プールを使用] または [固定 IP アドレスのリストを使用] を選択します。 [固定 IP アドレスのリストを使用] を選択した場合は、IP アドレス、ゲートウェイ、およびサブネット マスクのコンマ区切りのリストを指定する必要があります。
IP アドレス プール	IP 割り当てに [IP アドレス プールを使用] を選択した場合は、IP アドレス プール名を指定します。
物理 NIC	物理 NIC として vmnicX を使用するホスト トランスポート ノードとは異なり、NSX Edge トランスポート ノードは fp-ethX を使用します。

- 7 (オプション) GET <https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>>
API 呼び出しを使用して、トランスポート ノードを確認します。

```
GET https://<nsx-mgr>/api/v1/transport-nodes/78a03020-a3db-44c4-a8fa-f68ad4be6a0c

{
  "resource_type": "TransportNode",
  "id": "78a03020-a3db-44c4-a8fa-f68ad4be6a0c",
  "display_name": "node-comp-01b",
  "transport_zone_endpoints": [
    {
      "transport_zone_id": "efd7f38f-c5da-437d-af03-ac598f82a9ec",
      "transport_zone_profile_ids": [
        {
          "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
          "resource_type": "BfdHealthMonitoringProfile"
        }
      ]
    }
  ],
  "host_switches": [
    {
      "host_switch_profile_ids": [
        {
          "value": "8abdb6c0-db83-4e69-8b99-6cd85bfcc61d",
          "key": "UplinkHostSwitchProfile"
        },
        {
          "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
          "key": "LldpHostSwitchProfile"
        }
      ],
      "host_switch_name": "overlay-hostswitch",
      "pnics": [
        {
          "device_name": "vmnic1",
          "uplink_name": "uplink-1"
        }
      ],
      "static_ip_pool_id": "c78ac522-2a50-43fe-816a-c459a210127e"
    }
  ],
  "node_id": "c551290a-f682-11e5-ae84-9f8726e1de65",
  "_create_time": 1459547122893,
  "_last_modified_user": "admin",
  "_last_modified_time": 1459547126740,
  "_create_user": "admin",
  "_revision": 1
}
```

- 8 (オプション) 状態の情報を確認するには、GET `https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>/status` API 呼び出しを使用します。

```
{
  "control_connection_status": {
    "degraded_count": 0,
    "down_count": 0,
    "up_count": 1,
    "status": "UP"
  },
  "tunnel_status": {
    "down_count": 0,
    "up_count": 0,
    "status": "UNKNOWN",
    "bfd_status": {
      "bfd_admin_down_count": 0,
      "bfd_up_count": 0,
      "bfd_init_count": 0,
      "bfd_down_count": 0
    },
    "bfd_diagnostic": {
      "echo_function_failed_count": 0,
      "no_diagnostic_count": 0,
      "path_down_count": 0,
      "administratively_down_count": 0,
      "control_detection_time_expired_count": 0,
      "forwarding_plane_reset_count": 0,
      "reverse_concatenated_path_down_count": 0,
      "neighbor_signaled_session_down_count": 0,
      "concatenated_path_down_count": 0
    }
  },
  "pnic_status": {
    "degraded_count": 0,
    "down_count": 0,
    "up_count": 4,
    "status": "UP"
  },
  "mgmt_connection_status": "UP",
  "node_uuid": "cd4a8501-0ffc-44cf-99cd-55980d3d8aa6",
  "status": "UNKNOWN"
}
```

次のステップ

NSX Edge ノードを NSX Edge クラスタに追加します。「[「NSX Edge クラスタの作成」](#)」を参照してください。

NSX Edge クラスタの作成

NSX Edge のマルチノードクラスタがあると、1 つ以上の NSX Edge が常に使用可能になります。NAT やロード バランサなどのステートフル サービスを使用して Tier-0 論理ルーターまたは Tier-1 ルーターを作成するには、それを NSX Edge クラスタに関連付ける必要があります。そのため、NSX Edge が 1 つしかない場合でも、NSX Edge クラスタに属する必要があります。

1 台の NSX Edge トランスポート ノードは 1 つの NSX Edge クラスタにのみ追加できます。

1 つの NSX Edge クラスタを使用して複数の論理ルーターをバッキングできます。

NSX Edge クラスタの作成した後、これを編集して NSX Edge を追加できます。

前提条件

- 1 台以上の NSX Edge ノードを追加します。
- NSX Edge を管理プレーンに追加します。
- NSX Edge をトランスポート ノードとして追加します。
- オプションで、高可用性 (HA) 用に NSX Edge クラスタ プロファイルを作成します ([ファブリック] > [プロファイル] > [Edge クラスタ プロファイル])。デフォルトの NSX Edge クラスタ プロファイルを使用することもできます。

手順

- 1 ブラウザから、NSX Manager(<https://<nsx-manager-ip-address>>) に管理者権限でログインします。
- 2 [ファブリック] - [ノード] - [Edge クラスタ] - [追加] の順に移動します。
- 3 NSX Edge クラスタ名を入力します。
- 4 NSX Edge クラスタ プロファイルを選択します。
- 5 [編集] をクリックし、[物理マシン] または [仮想マシン] を選択します。
「物理マシン」は、ベアメタル上にインストールされている NSX Edge を意味します。「仮想マシン」は、仮想マシン/アプライアンスとして配置されている NSX Edge を意味します。
- 6 仮想マシンの場合、[メンバーのタイプ] ドロップダウン メニューから NSX Edge ノードまたは [Public Cloud Gateway ノード] のいずれかを選択します。
仮想マシンがパブリック クラウド環境に展開されている場合、Public Cloud Gateway を選択します。それ以外の場合には、NSX Edge ノードを選択します。
- 7 [使用可能] 列から NSX Edge を選択し、右矢印をクリックして [選択済み] 列に移動します。

次のステップ

これで、論理ネットワーク トポロジを構築してサービスを設定できるようになります。『NSX-T 管理ガイド』を参照してください。

NSX Cloud コンポーネントのインストール

9

NSX Cloud には、複数のパブリック クラウド ネットワークを 1 つの画面で管理するための機能を提供します。

NSX Cloud は、プロバイダ固有のネットワークに依存せず、パブリック クラウドでのハイパーバイザーのアクセスを必要としません。

また、次のようなメリットがあります。

- 本番環境で使用するものと同じネットワーク プロファイルやセキュリティ プロファイルを使用して、アプリケーションの開発およびテストを実施できます。
- 開発者は、開発中のアプリケーションを開発が終わるまで管理できます。
- ディザスタ リカバリによって、計画外の停止や、パブリック クラウドで発生したセキュリティの脅威からリカバリできます。
- パブリック クラウド間でワークロードを移行する場合、NSX Cloud では、移行先がどこであっても、ワークロード仮想マシンに類似のセキュリティ ポリシーを確実に適用できます。

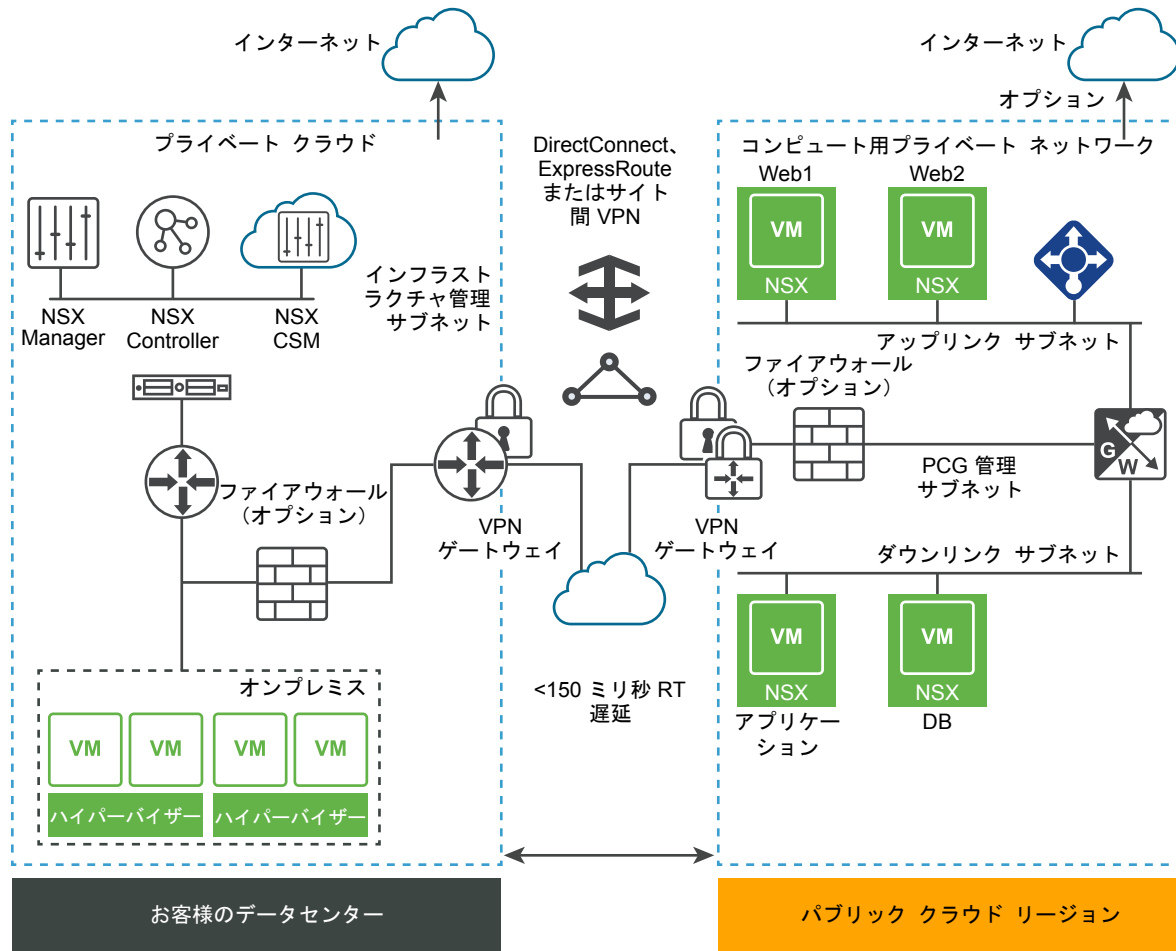
この章には、次のトピックが含まれています。

- [NSX Cloud のアーキテクチャとコンポーネント](#)
- [NSX Cloud コンポーネントのインストール](#)

NSX Cloud のアーキテクチャとコンポーネント

NSX Cloud は、NSX-T コア コンポーネント、NSX Manager、および NSX Controller をパブリック クラウドに組み込むことで、実装環境全体にネットワークとセキュリティを提供します。

図 9-1. NSX Cloud アーキテクチャ



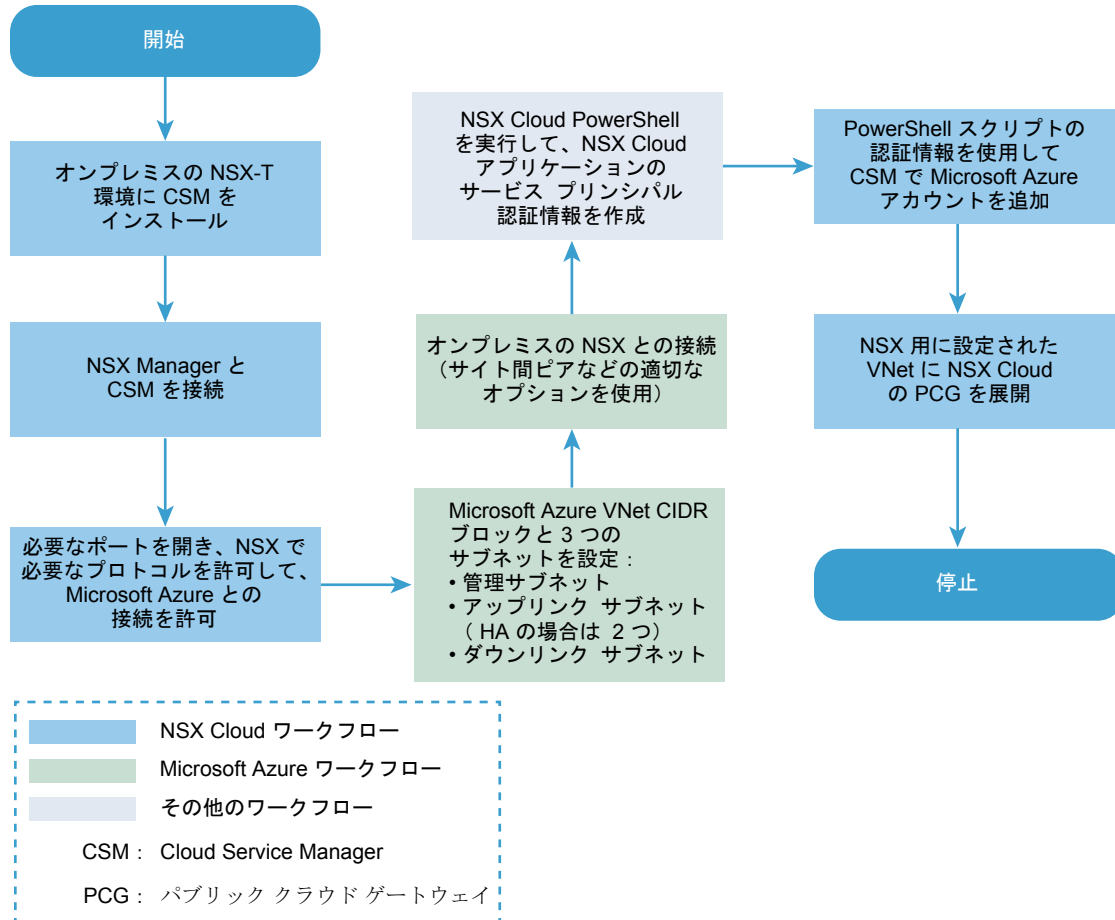
NSX Cloud の主要なコンポーネントは次のとおりです。

- NSX Manager : ロール ベースのアクセス コントロール (RBAC) が定義された管理プレーン。
- NSX Controller : 制御プレーンと実行時状態。
- Cloud Service Manager : NSX Manager に組み込むことで、管理プレーンにパブリック クラウド固有の情報を提供できます。
- NSX Public Cloud Gateway : NSX の管理プレーンと制御プレーン、NSX Edge Gateway Service との接続、およびパブリック クラウド エンティティとの API ベースの通信を提供。
- NSX Agent : ワークロード仮想マシンに NSX が管理するデータパスを提供。

NSX Cloud コンポーネントのインストール

このフローチャートは、パブリック クラウド内のワークロード仮想マシンを NSX-T で管理するための準備の概要を示しています。

図 9-2. NSX Cloud の準備ワークフロー



CSM のインストールと NSX Manager との接続

CSM のインストール

Cloud Service Manager (CSM) は、NSX Cloud の重要なコンポーネントです。

コア NSX-T コンポーネントのインストール後に、CSM をインストールします。

詳細については、[「NSX Manager および利用可能なアプライアンスのインストール」](#) を参照してください。

NSX Manager の FQDN の公開

NSX-T のコア コンポーネントと CSM をインストールした後、完全修飾ドメイン名 (FQDN) を使用して NAT を有効にするには、環境内の NSX-T DNS サーバに、正引きと逆引きのエントリを設定します。

また、NSX-T API を使用して NSX Manager の FQDN を公開できるようにする必要があります。

要求の例：[PUT https://<nsx-mgr>/api/v1/configs/management]

```
{
  ["publish_fqdns": true],
  "_revision": 0
}
```

応答の例：

```
{
  "publish_fqdns": true,
  "_revision": 1
}
```

詳細については、『NSX-T API ガイド』を参照してください。

NSX Manager への CSM の追加

CSM アプライアンスと NSX Manager を接続して、これらのコンポーネントが相互通信できるようにする必要があります。

前提条件

- NSX Manager がインストール済みで、NSX Manager にログインするための管理者権限が付与されている必要があります。
- CSM がインストール済みで、エンタープライズ管理者ロールが CSM に割り当てられている必要があります。

手順

- 1 NSX Manager への SSH セッションを開きます。
- 2 NSX Manager で **get certificate api thumbprint** コマンドを実行します。

```
NSX-Manager> get certificate api thumbprint
```

コマンドの出力は、この NSX Manager に固有の一連の数値です。

- 3 エンタープライズ管理者ロールで CSM にログインします。
- 4 [設定 (Settings)] - [NSX Manager の関連付け (Associate NSX Manager)] の順にクリックします。
- 5 NSX Manager の詳細を入力します。

オプション	説明
NSX Manager のホスト名	NSX Manager の完全修飾ドメイン名 (FQDN) を入力します (分かっている場合)。NSX Manager の IP アドレスを入力することもできます。
管理者認証情報	エンタープライズ管理者ロールを持ったユーザー名とパスワードを入力します。
NSX Manager のサムプリント	手順 2 で取得した NSX Manager のサムプリント値を入力します。

6 [接続 (Connect)] をクリックします。

CSM は、NSX Manager のサムプリントを確認して、接続を確立します。

次のステップ

パブリック クラウド アカウントをオンプレミス NSX-T 環境に接続します。

オンプレミスの NSX-T 環境と Microsoft Azure ネットワークとの接続

Microsoft Azure ネットワークとオンプレミスの NSX-T アプライアンス間に接続を確立する必要があります。

次は、Microsoft Azure ネットワークとオンプレミスの NSX-T 環境の接続の概要です。

注: NSX Manager がインストールされ、オンプレミス環境内の CSM と接続されている必要があります。

- 必要なネットワーク ポートを開き、Microsoft Azure の接続に必要なプロトコルを NSX Manager で許可します。
- 必要な CIDR ブロックと、NSX Manager で必要な 4 つのサブネットを使用して、Microsoft Azure VNet を設定します。
- 環境に最適なオプションを使用して、Microsoft Azure VNet とオンプレミスの NSX Manager アプライアンスの接続を設定します。
- CSM アプライアンスの時刻を、Microsoft Azure Storage サーバまたは NTP と同期させます。

パブリック クラウドから NSX Manager へのアクセス許可

次のネットワーク ポートおよびプロトコルを有効にして、オンプレミスの NSX Manager 環境との接続を可能にします。

表 9-1.

送信元	宛先	プロトコル/ポート	説明
PCG	NSX Manager	TCP/5671	パブリック クラウドからオンプレミス NSX-T への受信トラフィック (管理プレーンの通信用)。
PCG	NSX Manager	TCP/8080	パブリック クラウドからオンプレミス NSX-T への受信トラフィック (アップグレード用)。
PCG	NSX Controller	TCP/1234, TCP/1235	パブリック クラウドからオンプレミス NSX-T への受信トラフィック (制御プレーンの通信用)。
PCG	DNS	UDP/53	パブリック クラウドからオンプレミスの NSX-T DNS への受信トラフィック (オンプレミスの DNS サーバを使用している場合)。
CSM	PCG	TCP/7442	CSM 設定のプッシュ

表 9-1. (続き)

送信元	宛先	プロトコル/ポート	説明
任意	NSX Manager	TCP/443	NSX Manager ユーザー インターフェイス
任意	CSM	TCP/443	CSM ユーザー インターフェイス

重要: すべての NSX-T インフラストラクチャ通信で、SSL ベースの暗号化が利用されます。ファイアウォールで、非標準ポートを経由する SSL トラフィックが許可されるようにします。

Microsoft Azure VNet のセットアップ

Microsoft Azure で、ルーティング可能な CIDR ブロックを作成し、必要なサブネットを設定します。

- 1 つの管理サブネット。推奨範囲は /28 以上で次のトラフィックに使用します。
 - オンプレミス アプライアンスへの制御トラフィック
 - クラウド プロバイダ API エンドポイントへの API トラフィック
- 1 つのダウンリンク サブネット。推奨範囲は /24 で、ワークロードワークロード仮想マシンに使用します。
- 1 つのアップリンク サブネット (HA の場合は 2 つ)。推奨範囲は /24 で、VNet で送受信される North-South トラフィックのルーティングに使用します。

Microsoft Azure VNet とオンプレミスの NSX-T との接続

すべてのパブリック クラウドに、オンプレミス環境に接続するためのオプションが提供されています。要件に合わせて、使用可能な接続オプションのいずれかを選択できます。詳細については、[Azure のリファレンス ドキュメント](#)を参照してください。

注: Microsoft Azure によるセキュリティ上の考慮事項とベスト プラクティスを確認して実装する必要があります。たとえば、Microsoft Azure ポータルまたは API へのアクセス権を持つすべてのユーザー アカウントには、多要素認証 (MFA) を有効にする必要があります。多要素認証によって認証されたユーザーのみがポータルにアクセスできるようになると、認証情報が盗まれたり漏洩した場合でも、不正にアクセスされる可能性が低減します。詳細な情報および推奨事項については、[Azure Security Center のドキュメント](#)を参照してください。

NSX Cloud への Microsoft Azure サブスクリプションの追加

NSX Cloud に Microsoft Azure サブスクリプションを追加するには、Cloud Service Manager (CSM) に情報を提供する必要があります。

Microsoft Azure サブスクリプションには、テナント ID とサブスクリプション ID がすでに含まれていますが、サブスクリプションで NSX Cloud が動作するためには、新しいサービス プリンシパルを作成して必要なアクセス権を NSX に付与する必要があります。NSX Cloud には、サービス プリンシパルを生成するための PowerShell スクリプトが用意されています。

サービス プリンシパルの生成

NSX Cloud が Microsoft Azure サブスクリプションで動作するには、NSX-T に必要なアクセス権を付与するための新しいサービス プリンシパルを作成する必要があります。NSX Cloud には、サービス プリンシパルを生成するための PowerShell スクリプトが用意されています。

このセクションに記載されている PowerShell スクリプトを使用して、次を実行します。

- NSX Cloud 用の Azure Active Directory アプリケーションを作成します。
- NSX Cloud アプリケーションの Azure Resource Manager サービス プリンシパルを作成します。
- Microsoft Azure サブスクリプションを追加するために必要な認証情報を CSM で生成します。

前提条件

- サポート対象の Windows オペレーティングシステムに、PowerShell 5.0 以降と AzureRM モジュールをインストールしている必要があります。
- CSM がインストールされ、オンプレミスの NSX Manager と接続している必要があります。
- NSX Cloud のサービス プリンシパルを生成するにはスクリプトを実行する Microsoft Azure サブスクリプションの所有者権限が必要です。

手順

- 1 次の PowerShell スクリプトをコピーし、Windows デスクトップまたはサーバに保存します。
NSXCloud-Create-SP.ps1 という名前を推奨します。

注： このスクリプトを実行する前に、Microsoft Azure サブスクリプション ID を入力します。

```
# Input your Microsoft Azure Subscription ID. You must be the owner of this subscription.
$[subscriptionId ]= "<Your_Azure_Subscription_ID"

# You may change the location to store the output file generated from this script. The
default is C:\temp\
$outputFile = [string]::format("C:\temp\NSXCloud_ServicePrincipal_{0}.txt",
$subscriptionId)

# Log in to your Microsoft Azure subscription
$login= Login-AzureRMAccount
$login
$tenantId= (Get-AzureRmSubscription -SubscriptionId $subscriptionId).TenantId
Set-AzureRMContext -SubscriptionId $subscriptionId -TenantId $tenantId

# Create an NSX-Cloud Application in Active Directory
$key = [guid]::NewGuid().ToString()
Write-Output "Creating App Id "
$date = Get-date
$appIdUri = [string]::format("NSXCloudApp-{0}-{1}-{2}-{3}-{4}", $date.Year, $date.Month,
$date.Day, $date.Hour, $date.Minute)
```

```

Write-Output "App Id URI: $appIdUri"

Write-Output "Creating AAD application..."
$securePassword = ConvertTo-SecureString $key -AsPlainText -Force
$azureAdApplication = New-AzureRmADApplication -DisplayName $appIdUri -HomePage
"http://cloud.vmware.com/nsxcloud" -IdentifierUri $appIdUri -Password $securePassword
$azureAdApplication | Format-Table

# Create the Service Principal
Write-Output "Creating AAD Service Principal..."
$servicePrincipal = New-AzureRmADServicePrincipal -ApplicationId
$azureAdApplication.ApplicationId
$servicePrincipal | Format-Table

# Wait for the Service Principal to be created
Write-Output "Waiting for 10 seconds to allow the Service Principal to complete
generating..."
Start-Sleep -s 10

# Assign the Service Principal the Contributor Role in the Subscription.
# Roles can be granted at the Resource Group Level if desired.
Write-Output "Assigning the Contributor role to the service principal..."
New-AzureRmRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName
$azureAdApplication.ApplicationId

# The Application ID ( a.k.a. Client ID) will be required when creating the account in
the NSX-Cloud Service Principal
$message = [string]::format("[Azure Resource Manager Service Principal for NSX-
Cloud]")
Write-Output $message
$message|Out-File $outputFile -Append

$message = [string]::format("Client ID:{0}", $
($azureAdApplication.ApplicationId))
Write-Output $message
$message|Out-File $outputFile -Append

$message = [string]::format("Client Key:{0}", $key)
Write-Output $message
$message|Out-File $outputFile -Append

$message = [string]::format("Tenant ID: {0}", $tenantId)
Write-Output $message
$message|Out-File $outputFile -Append

$message = [string]::format("Subscription ID: {0}", $subscriptionId)
Write-Output $message
$message|Out-File $outputFile -Append

```

2 スクリプトを実行します。

- 3 Microsoft Azure 認証情報を入力して、Microsoft Azure サブスクリプションにログインします。

注：スクリプトを正常に実行するには、この Microsoft Azure サブスクリプションの所有者権限が必要です。

スクリプトによって NSX Cloud のサービス プリンシパルが生成されます。出力は、デフォルトで [C:\temp\] に保存されます。出力ファイル名は [NSXCloud_ServicePrincipal_<Your_Subscription_ID>.txt] になります。

注： 必要な場合は、デフォルトの出力場所と出力ファイル名をスクリプト内で変更できます。

- 4 出力ファイルに、次の認証情報が記載されていることを確認します。これらは、CSM に Microsoft Azure サブスクリプションを追加するために必要になります。
 - クライアント ID
 - クライアント キー
 - テナント ID
 - サブスクリプション キー
- 5 Microsoft Azure サブスクリプションに、[コントリビュータ (Contributor)] のロールで NSX Cloud アプリケーションが生成されたことを確認します。

[NSX Cloud のサービス プリンシパルを使用した CSM への Microsoft Azure サブスクリプションの追加]

NSX Cloud のサービス プリンシパルを使用した CSM への Microsoft Azure サブスクリプションの追加

NSX Cloud のサービス プリンシパルの詳細情報を入手したら、CSM に Microsoft Azure サブスクリプションを追加できます。

前提条件

- NSX-T のエンタープライズ管理者ロールが必要です。
- NSX Cloud のサービス プリンシパルの詳細情報が含まれる PowerShell スクリプトの出力が必要です。

手順

- 1 エンタープライズ管理者ロールを持つアカウントを使用して、CSM にログインします。
- 2 [CSM] - [クラウド (Clouds)] - [Azure] の順に移動します。
- 3 [+ (追加) (+Add)] マークをクリックし、次の詳細を入力します。

オプション	説明
名前	CSM で、アカウントを識別するための適切な名前を指定します。1 つの Microsoft Azure テナント ID に、複数の Microsoft Azure サブスクリプションが関連付けられている場合があります。アカウント名に「Account」を含めると、CSM 内で分かりやすい名前になります。たとえば、Azure-DevOps-Account や Azure-Finance-Account などにします。
クライアント ID	PowerShell スクリプトの出力からこの値をコピーして、貼り付けます。
テナント ID	PowerShell スクリプトの出力からこの値をコピーして、貼り付けます。

オプション	説明
サブスクリプション ID	PowerShell スクリプトの出力からこの値をコピーして、貼り付けます。
キー	PowerShell スクリプトの出力からこの値をコピーして、貼り付けます。

- 4 [保存 (Save)] をクリックします。

CSM でアカウントが追加されて、数分以内に [アカウント (Accounts)] セクションに表示されます。

次のステップ

[\[Microsoft Azure VNet への PCG の展開\]](#)

PCG の展開

NSX Public Cloud Gateway (PCG) はパブリック クラウドに展開すると、NSX-T を使用してパブリック クラウド イベントトリを管理できるようになります。

Microsoft Azure VNet への PCG の展開

NSX Public Cloud Gateway (PCG) は、パブリック クラウドと NSX-T オンプレミス管理コンポーネント間の North-South 接続を可能にします。

PCG 環境は、NSX コンポーネントの完全修飾ドメイン名 (FQDN) と、FQDN を解決する DNS サーバを使用した、既存のネットワーク アドレス設定に適応します。

注: PCG を使用する場合、パブリック クラウドと NSX-T との接続に IP アドレスを使用することは推奨されませんが、この方法を選択した場合は、IP アドレスを変更しないでください。

前提条件

- Microsoft Azure サブスクリプションが、CSM に追加されている必要があります。
- PCG を展開する Microsoft Azure VNet に、アップリンク、ダウンリンク、および管理用の 3 つのサブネットが必要です (HA を有効にする場合は 4 つ)。

手順

- 1 エンタープライズ管理者ロールを持つアカウントを使用して、CSM にログインします。
- 2 [クラウド (Clouds)] - [Azure] をクリックし、[VNet] タブに移動します。
- 3 PCG を展開する VNet をクリックします。
- 4 [ゲートウェイの展開] をクリックします。[PCG の展開] ウィザードが表示されます。

5 一般的なプロパティについては、次のガイドラインを考慮します。

オプション	説明
SSH パブリック キー	PCG の展開で検証するための SSH パブリック キーを指定します。これは、PCG の展開ごとに必要です。
検疫ポリシー	PCG を初めて展開する場合は、デフォルトの[無効 (disabled)]モードのままにします。この値は、仮想マシンのオンボーディング後に変更できます。詳細については、『NSX-T 管理ガイド』の「[検疫ポリシーの管理 (Manage Quarantine Policy)]」を参照してください。
ローカル ストレージ アカウント	CSM に Microsoft Azure サブスクリプションを追加すると、Microsoft Azure ストレージアカウントのリストが CSM で使用できるようになります。ドロップダウン メニューからストレージ アカウントを選択します。PCG の展開で CSM は、パブリックに使用可能な PCG の仮想ハードディスク (VHD) を、選択したリージョンのストレージ アカウントにコピーします。 注： 前回の PCG の展開で、仮想ハードディスク イメージをリージョン内の該当のストレージ アカウントにコピーしている場合、以降の展開では、この場所のイメージが使用して展開時間を短縮します。
または 仮想ハードディスクの URL	公開されている VMware のリポジトリで提供されない別の PCG イメージを使用する場合は、PCG の仮想ハードディスクの URL をここに入力できます。仮想ハードディスクは、この VNet が作成された同じアカウントと同じリージョンに配置されている必要があります。
詳細	DNS の詳細設定を使用すると、NSX-T 管理コンポーネントを解決するための DNS サーバを柔軟に選択できます。
パブリック クラウド プロバイダを DHCP 経由で取得	Microsoft Azure の DNS 設定を使用する場合は、このオプションを選択します。DNS 設定を上書きするオプションを選択していない場合は、これがデフォルトの DNS 設定になります。
パブリック クラウド プロバイダの DNS サーバ情報の変更	1 台または複数の DNS サーバの IP アドレスを手動で指定して、NSX-T アプライアンスと、この VNet 内のワークロード仮想マシンを解決する場合は、このオプションを選択します。
NSX-T アプライアンスにのみパブリック クラウド プロバイダの DNS サーバを使用	Microsoft Azure の DNS サーバを使用して NSX-T 管理コンポーネントを解決するには、このオプションを選択します。この設定では、2 台の DNS サーバを使用できます。1 台は NSX-T アプライアンスを解決する PCG 用で、もう 1 台は、この VNet 内のワークロード仮想マシンを解決する VNet 用です。

6 [次へ (Next)] をクリックします。

7 [サブネット (Subnets)] では、次のガイドラインを考慮します。

オプション	説明
NSX クラウド ゲートウェイの HA の有効化	高可用性を有効にするには、このオプションを選択します。
サブネット	高可用性を有効にするには、このオプションを選択します。
管理 NIC のパブリック IP アドレス	[新しい IP アドレスの割り当て (Allocate New IP address)] を選択して、管理 NIC にパブリック IP アドレスを指定します。Azure VNet で空いているパブリック IP アドレスを再利用する場合は、パブリック IP アドレスを手動で指定できます。

PCG の展開が正常に完了すると、以下のエンティティが NSX Manager に自動的に作成されます。

- Public Cloud Gateway (PCG) という名前の Edge ノードが作成されます。
- PCG は Edge クラスタに追加されます。
- PCG は、2 つのトランスポート ゾーンが作成されたトランスポート ノードとして登録されます。
- デフォルトの論理スイッチが 2 台作成されます。

- Tier-0 論理ルーターが 1 つ作成されます。
- IP アドレス検出プロファイルが作成されます。これは、オーバーレイ論理スイッチで使われます。
- DHCP プロファイルが作成されます。これは、DHCP サーバで使われます。
- デフォルトの NSGroup が PublicCloudSecurityGroup という名前で作成されます。これには、次のメンバーが含まれます。
 - デフォルトの VLAN 論理スイッチ。
 - 論理ポート (PCG アップリンク ポートに 1 つずつ、高可用性が有効な場合)。
 - IP アドレス
- デフォルトの分散ファイアウォール (DFW) ルールが 3 つ作成されます。
 - LogicalSwitchToLogicalSwitch
 - LogicalSwitchToAnywhere
 - AnywhereToLogicalSwitch

次のステップ

ワークロード仮想マシンをオンボーディングします。Day-N ワークフローについては、NSX-T 管理ガイドの「[[ワークロード仮想マシンのオンボーディングと管理 (Onboarding and Managing Workload VMs)]]」を参照してください。

PCG の展開の解除

PCG の展開を解除するには、まず、関連付けられている作成済みの論理エンティティをすべて削除する必要があります。

PCG の展開の解除

PCG の展開を解除するには、CSM で [クラウド (Clouds)] - [Azure] - [VNet] の順に選択し、[ゲートウェイの展開解除] をクリックします。

PCG の展開が解除されると、NSX Cloud によって作成されたデフォルト エンティティは自動的に削除されます。

ゲートウェイの展開を解除する前に、NSX Manager で作成したすべての論理エンティティを削除します。削除するエンティティを見つけるには、以下のリストを参照してください。

- パブリック クラウドの DNS エントリ
- DDI : DHCP プロファイル
- ルーティング : SNAT ルール
- ルーティング : 静的ルーター
- ルーティング : 論理ルーター ポート
- ルーティング : 論理ルーター
- ファブリック ノード : Edge クラスタ

- ファブリック ノード： トランスポート ノード
- ファブリック ノード： Edge
- ファブリック プロファイル： PCG-Uplink-HostSwitch-Profile
- スイッチング： 論理スイッチ ポート
- スイッチング： 論理スイッチ
- ファブリック トランスポート ゾーン： トランスポート ゾーン
- スイッチング： PublicCloud-Global-SpoofGuardProfile

NSX-T のアンインストール

NSX-T オーバーレイの要素の削除、NSX-T からのハイパーバイザー ホストの削除、NSX-T の完全なアンインストールが可能です。

この章には、次のトピックが含まれています。

- [NSX-T オーバーレイの設定解除](#)
- [NSX-T からのホストの削除、または NSX-T の完全なアンインストール](#)

NSX-T オーバーレイの設定解除

オーバーレイは削除するがトランスポート ノードは残す場合、次の手順を実行します。

手順

- 1 仮想マシン管理ツールを使用して、すべての論理スイッチからすべての仮想マシンを接続解除します。
- 2 NSX Manager のユーザー インターフェイスまたは API で、すべての分散論理ルーターを削除します。
- 3 NSX Manager のユーザー インターフェイスまたは API で、すべての論理スイッチ ポートを削除し、すべての論理スイッチを削除します。
- 4 NSX Manager のユーザー インターフェイスまたは API で、すべての NSX Edge を削除し、すべての NSX Edge クラスタを削除します。
- 5 必要に応じて、新しい NSX-T オーバーレイを設定します。

NSX-T からのホストの削除、または NSX-T の完全なアンインストール

NSX-T を完全にアンインストールするか、NSX-T からハイパーバイザー ホストを削除して NSX-T オーバーレイでホストが動作しないようにするには、次の手順を実行します。

次の手順で、NSX-T のクリーン アンインストールを実行します。

前提条件

仮想マシン管理ツールが vCenter Server の場合は、vSphere ホストをメンテナンス モードに切り替えます。

手順

- 1 仮想マシン管理ツールで、ホスト上のすべての仮想マシンを NSX-T 論理スイッチから接続解除します。

- 2 NSX Manager ユーザー インターフェイスの [ファブリック (Fabric)] > [ノード (Nodes)] > [トランスポート ノード (Transport Nodes)] から、または **DELETE /api/v1/transport-node/<node-id>** API を使用して、ホストのトランスポート ノードを削除します。

トランスポート ノードを削除すると、ホストから N-VDS が削除されます。これは、次のコマンドを実行して確認できます。

```
[root@host:~] esxcli network vswitch dvs vmware list
```

KVM の場合、コマンドは次のようになります。

```
ovs-vsctl show
```

- 3 NSX Manager の CLI で、NSX-T install-upgrade サービスが実行されていることを確認します。

```
nsx-manager-1> get service install-upgrade
Service name: install-upgrade
Service state: running
Enabled: True
```

- 4 管理プレーンからホストをアンインストールして、NSX-T モジュールを削除します。

すべての NSX-T モジュールが削除されるまで、最大で 5 分程度かかる場合があります。

NSX-T モジュールを削除する方法は複数あります。

- NSX Manager で、[ファブリック (Fabric)] - [ノード (Nodes)] - [ホスト (Hosts)] - [削除 (Delete)] の順に選択します。

[NSX コンポーネントのアンインストール (Uninstall NSX Components)] が選択されていることを確認します。これによって NSX-T モジュールがホストからアンインストールされます。

RHEL 7.4 と依存関係にあるパッケージ、json_spirit、python-greenlet、libev、protobuf、leveldb、python-gevent、python-simplejson、glog を削除します。

Ubuntu 16.04.x と依存関係にあるパッケージ、nicira-ovs-hypervisor-node、openvswitch-switch、openvswitch-datapath-dkms、openvswitch-pki、python-openvswitch、openvswitch-common、libjson-spirit を削除します。

[ファブリック (Fabric)] - [ノード (Nodes)] - [ホスト (Hosts)] - [削除 (Delete)] の順に選択しても、[NSX コンポーネントのアンインストール (Uninstall NSX Components)] オプションが選択されていないと、ホストの登録は解除されません。ホストの状態に問題がある場合は、回避策として、このオプションを選択解除します。

- (コンピュート マネージャによって管理されるホスト) NSX Manager で、[ファブリック (Fabric)] - [ノード (Nodes)] - [ホスト (Hosts)] - [トランスポート ノード (Transport Nodes)] - [ホストの削除 (Delete Host)] の順に選択します。

NSX Manager で、[ファブリック (Fabric)] - [ノード (Nodes)] - [ホスト (Hosts)] - [コンピュート マネージャ (Compute Manager)] - [クラスタ マネージャの設定 (Configure Cluster Manager)] の順に選択し、[NSX を自動的にインストール (Automatically Install NSX)] を選択解除します。ノードを選択し、[NSX のアンインストール (Uninstall NSX)] をクリックします。

[NSX コンポーネントのアンインストール (Uninstall NSX Components)] が選択されていることを確認します。これによって NSX-T モジュールがホストからアンインストールされます。

- **DELETE /api/v1/fabric/nodes/<node-id>** API を使用します。

注: この API では、nsx-lcp バンドルから依存関係にあるパッケージは削除されません。

RHEL 7.4 と依存関係にあるパッケージ、json_spirit、python-greenlet、libev、protobuf、leveldb、python-gevent、python-simplejson、glog を削除します。

Ubuntu 16.04.x と依存関係にあるパッケージ、nicira-ovs-hypervisor-node、openvswitch-switch、openvswitch-datapath-dkms、openvswitch-pki、python-openvswitch、openvswitch-common、libjson-spirit を削除します。

- vSphere の CLI を使用します。

- a 管理用のサムプリントを取得します。

```
manager> get certificate api thumbprint
```

- b ホストの NSX-T CLI で次のコマンドを実行し、管理プレーンからホストを接続解除します。

```
host> detach management-plane <MANAGER> username <ADMIN-USER> password  
<ADMIN-PASSWORD> thumbprint <MANAGER-THUMBPRINT>
```

- c ホストで次のコマンドを実行し、フィルタを削除します。

```
[root@host:~] vsipioctl clearallfilters
```

- d ホストで次のコマンドを実行し、netcpa を停止します。

```
[root@host:~] /etc/init.d/netcpad stop
```

- e ホスト上の仮想マシンをパワーオフするか、別のホストに移行します。

- f ホストから NSX-T モジュールを手動でアンインストールします。

注: モジュールを個別に削除することはできません。1 回のコマンドですべてのモジュールを削除する必要があります。

```
esxcli software vib remove -n nsx-shared-libs -n nsx-common-libs -n nsx-metrics-  
libs -n nsx-rpc-libs -n nsx-nestdb-libs -n nsxa -n nsx-lldp -n nsx-da -n nsx-  
exporter -n nsx-aggservice -n nsxcli -n nsx-python-protobuf -n nsx-sfhc -n nsx-  
netcpa -n nsx-mpa -n nsx-esx-datapath -n nsx-host -n nsx-support-bundle-client -n  
nsx-nestdb -n nsx-platform-client -n nsx-hyperbus -n nsx-ctxteng -n nsx-python-  
gevent -n nsx-python-greenlet
```

- RHEL 7.4 では、**sudo yum remove nsx* <package-name>** コマンドを使用します。

依存関係にあるパッケージ、glog、json_spirit、kmod-openvswitch、nicira-ovs-hypervisor-node、openvswitch、openvswitch-selinux-policy、python-simplejson を削除します。

- Ubuntu 16.04.x では、**apt-get remove "nsx*" <package-name>** コマンドを使用します。

依存関係にあるパッケージ、nicira-ovs-hypervisor-node、openvswitch-switch、openvswitch-datapath-dkms、openvswitch-pki、python-openvswitch、openvswitch-common、libjson-spirit を削除します。

次のステップ

この変更を行うと、ホストが管理プレーンから削除され、NSX-T オーバーレイに含まれなくなります。

NSX-T を完全に削除する場合、仮想マシン管理ツールで NSX Manager、NSX Controller、および NSX Edge をシャットダウンしてディスクから削除します。