

NSX-T インストール ガイド

VMware NSX-T Data Center 1.1



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 © 2017 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

- データ プレーン 8
- 制御プレーン 8
- 管理プレーン 8
- NSX Manager 9
- NSX Controller 10
- 論理スイッチ 10
- 分散論理ルーター 10
- NSX Edge 11
- トランスポート ゾーン 12
- 主な概念 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 19
- NSX Manager が使用する TCP および UDP ポート 20
- NSX Controller によって使用される TCP および UDP ポート 21
- NSX Edge が使用する TCP および UDP ポート 22
- Key Manager が使用する TCP ポート 23
- インストールの概要 24

3 KVM の使用 26

- KVM のセットアップ 26
- KVM CLI を使用したゲスト仮想マシンの管理 30

4 NSX Manager のインストール 32

- vSphere Web Client を使用した ESXi への NSX Manager のインストール 33
- 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 36
- KVM への NSX Manager のインストール 39

5 NSX Controller のインストールとクラスタリング 42

- GUI を使用した ESXi への NSX Controller のインストール 43
- 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 46
- KVM への NSX Controller のインストール 49
- NSX Controller の管理プレーンへの追加 51
- 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 53

[クラスター マスターを使用した NSX Controller の追加](#) 54

6 NSX Edge のインストール 58

[NSX Edge のネットワーク設定](#) 59

[GUI を使用した ESXi への NSX Edge のインストール](#) 64

[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](#) 67

[PXE サーバで ISO ファイルを使用した NSX Edge のインストール](#) 71

[ベア メタルへの NSX Edge のインストール](#) 77

[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](#) 79

[NSX Edge の管理プレーンへの追加](#) 82

7 ホストの準備 84

[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](#) 84

[NSX-T ファブリックへのハイパーバイザー ホストの追加](#) 85

[NSX-T カーネル モジュールの手動インストール](#) 90

[ハイパーバイザー ホストの管理プレーンへの追加](#) 95

8 トランスポート ゾーンとトランスポート ノード 98

[トランスポート ゾーンについて](#) 98

[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](#) 100

[アップリンク プロファイルの作成](#) 103

[トランスポート ゾーンの作成](#) 106

[ホスト トランスポート ノードの作成](#) 108

[NSX Edge トランスポート ノードの作成](#) 114

[NSX Edge クラスターの作成](#) 118

9 NSX-T のアンインストール 120

[NSX-T オーバーレイの設定解除](#) 120

[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](#) 120

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 テクノロジーおよび仮想データセンターの運用に詳しい方向けに書かれています。本書は、VMware ESX、vCenter Server、および vSphere Web Client を含む VMware vSphere 5.5/6.0 や VMware OVF Tool などの仮想マシン管理サービス、またはカーネル ベースの仮想マシン (KVM) と連携する別の仮想マシン管理サービスについての知識があることを前提としています。

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。当社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

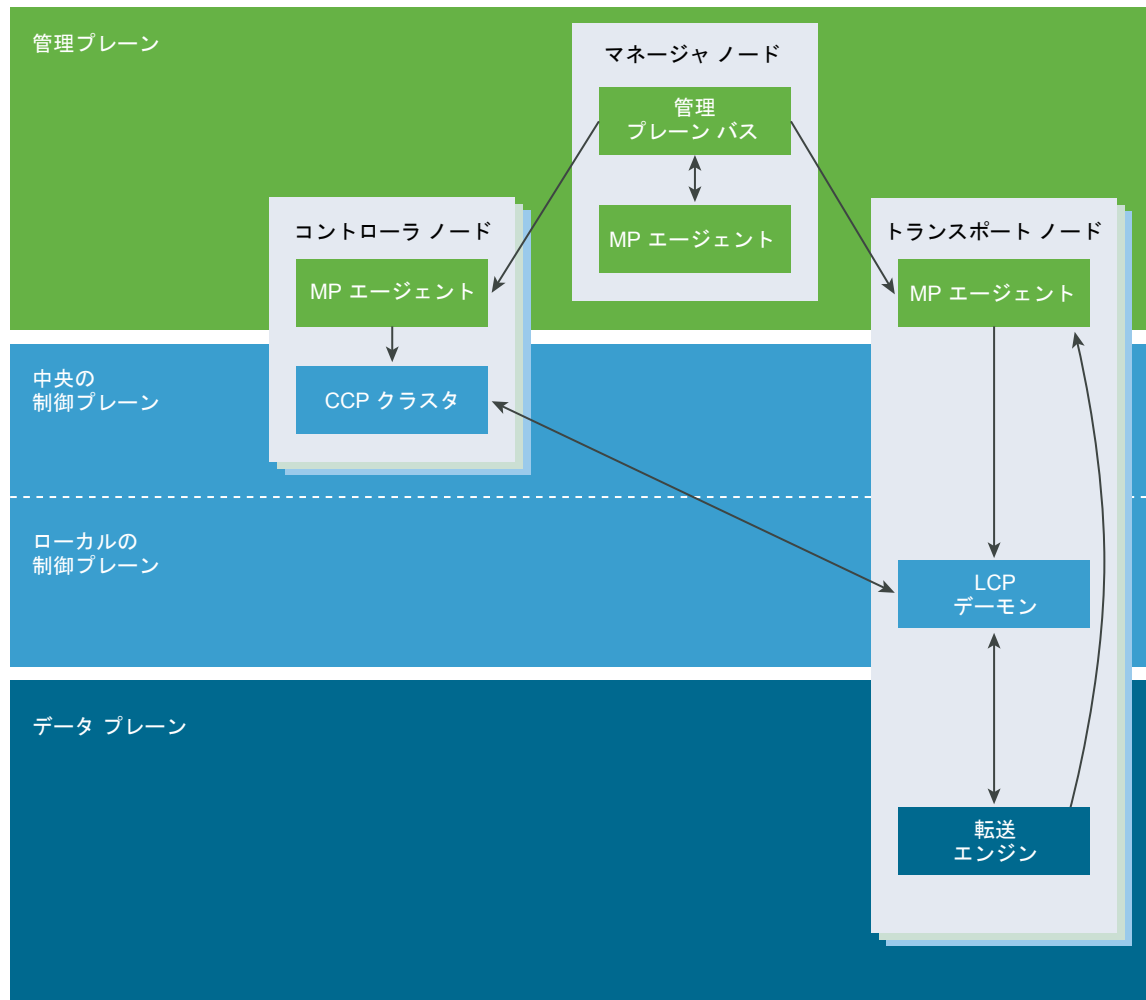
サーバ仮想化では、ソフトウェア ベースの仮想マシンの作成、削除、リストア、およびスナップショットの作成をプログラムによって行います。NSX-T のネットワーク仮想化は、ほぼ同じ方法で、ソフトウェア ベースの仮想ネットワークを作成、削除、リストア、およびスナップショットの作成を行います。

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、アクセス制御、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 インストールはそれぞれ単一の NSX Manager ノードをサポートし、NSX Manager クラスタをサポートしません。
- 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。

- データ プレーン
- 制御プレーン
- 管理プレーン
- NSX Manager
- NSX Controller
- 論理スイッチ
- 分散論理ルーター
- NSX Edge
- トランスポート ゾーン
- 主な概念

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。

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、ネットワークへのシグナル伝達と説明されることがあります。固定ユーザー構成がある場合に、データ プレーンをメンテナンスするためにメッセージを処理する際には、制御プレーンでその処理を行います。たとえば、仮想マシン (VM) の vMotion に応答するのは制御プレーンの役割ですが、仮想マシンを論理ネットワークに接続するのは管理プレーンの役割です。制御プレーンは、データ プレーン要素からのトポロジ情報のリフレクタとして機能することがよくあります (VTEP 用の MAC/トンネル マッピングなど)。その他の場合、制御プレーンはいくつかのデータ プレーン要素から受信したデータを処理して、いくつかのデータ プレーン要素を構成 (または再構成) します。たとえば、VIF ロケータを使用して、トンネルの正しいサブセットメッシュを計算し、確立します。

制御プレーンが処理するオブジェクトのセットには、VIF、論理ネットワーク、論理ポート、論理ルーター、IP アドレスなどが含まれます。

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 ステータスを監視し、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。

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、すべてのクラスとトランスポート 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。ユースケースの例として、中央の管理ノード アドレスの認証情報、パッケージ、統計情報、ステータスなどのブートストラッピング設定があります。MPA は制御プレーンとデータ プレーンから独立して実行でき、プロセスがクラッシュするか、反応しなくなった場合は独立して再起動できますが、同じホストで実行されているため運命をともにする場合もあります。MPA はローカル アクセスとリモート アクセスが可能です。MPA はトランスポート ノード、制御ノード、管理ノードで動作してノード管理を行います。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、コントローラ、論理スイッチ、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 を提供します。

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。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 Distributed Firewall：Edge サービスとセキュリティ サービスは、NSX Manager の組み込みコンポーネントとして、または連携するサードパーティ ベンダーから提供されます。

NSX Manager では、組み込みサービスと外部サービスとのシームレスな連携が可能です。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 Controller

NSX Controller は、仮想ネットワークとオーバーレイ転送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 の中央制御プレーン (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。

論理スイッチ

NSX Edge 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。

仮想データセンターのクラウド環境には、複数のテナントに跨るさまざまなアプリケーションが存在します。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。

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。

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。

分散論理ルーターは、従来型のネットワーク ハードウェア ルーターの中で設定が可能な部分です。ハードウェアの機能を複製し、単一のルーター内に複数のルーティングドメインを作成します。分散論理ルーターは物理ルーターで処理できるタスクの一部を実行します。また、それぞれ複数のルーティング インスタンスやルーティング テーブルを含めることができます。分散論理ルーターの使用は、ルーターの使用率を最大にする効果的な方法です。単一の物理ルーター内の複数の分散論理ルーターで、以前は複数の装置で実行していた処理を実行できるからです。

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すべて 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。

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 によって、複数のサブネットにわたっている同一ホスト上に存在する仮想マシンまたはワークロードは、従来のルーティング インターフェイスを経由することなく相互に通信できます。

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、論理スイッチでアクセスできるホストを制御します。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。

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。

ノードに 1 台以上のホストスイッチが含まれている場合、そのノードはトランスポート ノードとして機能できます。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 によってホストにホストスイッチ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、別のホストスイッチがインストールされます。ホストスイッチは、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 分散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。

主な概念

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 の一般的な概念について説明します。

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ノードです。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。

ファブリック プロファイル	NSX Edge クラスタに関連付けることができる特定の設定を表します。ファブリック プロファイルには、たとえば、停止したピアを検出するためのトンネリング プロパティを含めることができます。
論理ポート出力	仮想マシンまたは論理ネットワークへのネットワーク トラフィックを出力と呼びます。トラフィックがデータセンター ネットワークを離れ、仮想領域に入るためです。
論理ポート入力	仮想マシンからデータセンター ネットワークへのネットワーク トラフィックを入力と呼びます。トラフィックが物理ネットワークに入るためです。
分散論理ルーター	NSX-T のルーティング エンティティです。
分散論理ルーター 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 API 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ブロードキャスト 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に依存せず、多数のハイパーバイザーに跨る論理エンティティであり、物理的な場所を問わずに仮想マシンを接続します。このため、テナント ネットワークの管理者が再設定することなく仮想マシンを移行できます。</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。論理スイッチは分散論理ルーターを使用して接続でき、分散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分散論理ルーター インターフェイスへの接続を確立するための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報告します。
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操作を行います。管理プレーンは、ユーザー設定のクエリ、変更、維持も行います。
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	高可用性の監視に関わる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 アドレス ルーティングと IP アドレス サービスの機能のための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-T ホストスイッチまたは KVM Open vSwitch (OVS)

ハイパーバイザー上で実行され、物理的なトラフィック転送を行うソフトウェアです。ホストスイッチまたは OVS はテナント ネットワークの管理者から認識できず、各論理スイッチが依存する、基盤となる転送サービスを提供します。ネットワークを仮想化するには、ネットワーク コントローラが、テナントの管理者が論理スイッチを作成、設定したときに定義した論理ブロードキャスト ドメインを形成するネットワーク フロー テーブルを使用してハイパーバイザー ホストスイッチを設定する必要があります。

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、仮想マシンと分散論理ルーターの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、仮想マシンの作成、移動、削除に伴ってハイパーバイザー ホストスイッチのフロー テーブルが更新されることを確認します。

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。

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ハイパーバイザー ホスト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。NSX Edge のスイッチング コンポーネントは OVS に基づいています。

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 分散論理ルーター

プロバイダ分散論理ルーターは、物理ネットワークへの Tier-0 分散論理ルーターとも呼ばれます。Tier-0 分散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。分散論理ルーターは 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、分散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 分散論理ルーター

Tier-1 分散論理ルーターは、2 番目の分散論理ルーターです。north バウンド接続用に 1 台の Tier-0 分散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 分散論理ルーターには、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、同じように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。NSX-T は、論理スイッチで有効になっている機能がわかるため、必要なサポート ソフトウェア パッケージをホストに展開できます。

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 の間の接続を提供する、仮想マシン上のネットワーク 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。

VTEP

仮想トンネル エンドポイントです。トンネル エンドポイントによって、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 Manager が使用する TCP および UDP ポート
- NSX Controller によって使用される TCP および UDP ポート
- NSX Edge が使用する TCP および UDP ポート
- Key Manager が使用する TCP ポート
- インストールの概要

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 バージョンに関する固有の要件があります。

ハイパーバイザー

表 2-1. 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
ESXi	<ul style="list-style-type: none">■ 6.5■ 6.0 パッチ リリース P04	4	16 GB
RHEL KVM	7.1 (3.10.0-229 カーネルのみ)、7.2 (3.10.0-327 カーネルのみ)	4	16 GB
Ubuntu KVM	14.04.x (3.13 または 4.4 カーネル)、16.04.x (4.4 カーネルのみ)	4	16 GB

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** を実行する際にはカーネルのアップデートを無効にしてください。また、**yum install** の実行後に、カーネルのバージョンが NSX-T でサポートされていることを確認してください。

NSX Manager および NSX Controller

表 2-2. NSX Manager および 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
NSX Manager	16 GB	2	140 GB
NSX Controller	16 GB	2	120 GB

NSX Manager と NSX Controller の仮想マシンは、vSphere ESXi 5.5 以降でサポートされています。

NSX Edge

表 2-3. NSX Edge のリソース要件

展開規模	メモリ	vCPU	ディスク容量
小規模	4 GB	2	120 GB
中規模	8 GB	4	120 GB
大規模	16 GB	8	120 GB

表 2-4. NSX Edge の物理ハードウェア要件

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) ■ Xeon E5-xxxx (Sandy Bridge)
NIC	<ul style="list-style-type: none"> ■ Intel 82599 ■ Intel X540

ベア メタル NSX Edge のシステム要件

製品コード

- X520QDA1
- E10G42BT (X520-T2)
- E10G42BTDA (X520-DA2)
- E10G42BTDABLK
- X520DA1OCP

- X520DA2OCP
- E10G41BFSR (X520-SR1)
- E10G41BFSRBLK
- E10G42BFSR (X520-SR2)
- E10G42BFSRBLK
- E10G41BFLR (X520-LR1)
- E10G41BFLRBL

NIC PCI デバイス ID	説明
0x10F7	IXGBE_DEV_ID_82599_KX4
0x1514	IXGBE_DEV_ID_82599_KX4_MEZZ
0x1517	IXGBE_DEV_ID_82599_KR
0x10F8	IXGBE_DEV_ID_82599_COMBO_BACKPLANE
0x000C	IXGBE_SUBDEV_ID_82599_KX4_KR_MEZZ
0x10F9	IXGBE_DEV_ID_82599_CX4
0x10FB	IXGBE_DEV_ID_82599_SFP
0x11A9	IXGBE_SUBDEV_ID_82599_SFP
0x1F72	IXGBE_SUBDEV_ID_82599_RNDC
0x17D0	IXGBE_SUBDEV_ID_82599_560FLR
0x0470	IXGBE_SUBDEV_ID_82599_ECNA_DP
0x152A	IXGBE_DEV_ID_82599_BACKPLANE_FCOE
0x1529	IXGBE_DEV_ID_82599_SFP_FCOE
0x1507	IXGBE_DEV_ID_82599_SFP_EM
0x154D	IXGBE_DEV_ID_82599_SFP_SF2
0x154A	IXGBE_DEV_ID_82599_SFP_SF_QP
0x1558	IXGBE_DEV_ID_82599_QSFP_SF_QP
0x1557	IXGBE_DEV_ID_82599EN_SFP
0x10FC	IXGBE_DEV_ID_82599_XAUI_LOM
0x151C	IXGBE_DEV_ID_82599_T3_LOM
0x1528	IXGBE_DEV_ID_X540T
0x1560	IXGBE_DEV_ID_X540T1

NSX Manager のブラウザのサポート

表 2-5. NSX Manager のブラウザのサポート

ブラウザ	Windows 10	Windows 8.1	Windows 7	Ubuntu 12, 14.04	Max OSX 10.9, 10.10, 10.11
Internet Explorer 11		○	○		
Firefox 50		○	○	○	○
Chrome 54	○	○	○	○	○
Safari 9					○
Microsoft Edge 25	○				

ポートとプロトコル

次の図は、NSX-T 内のすべてのノード間通信パス、パスのセキュリティ確保と認証の方法、そして相互認証の確立に使用される認証情報の格納場所を示しています。

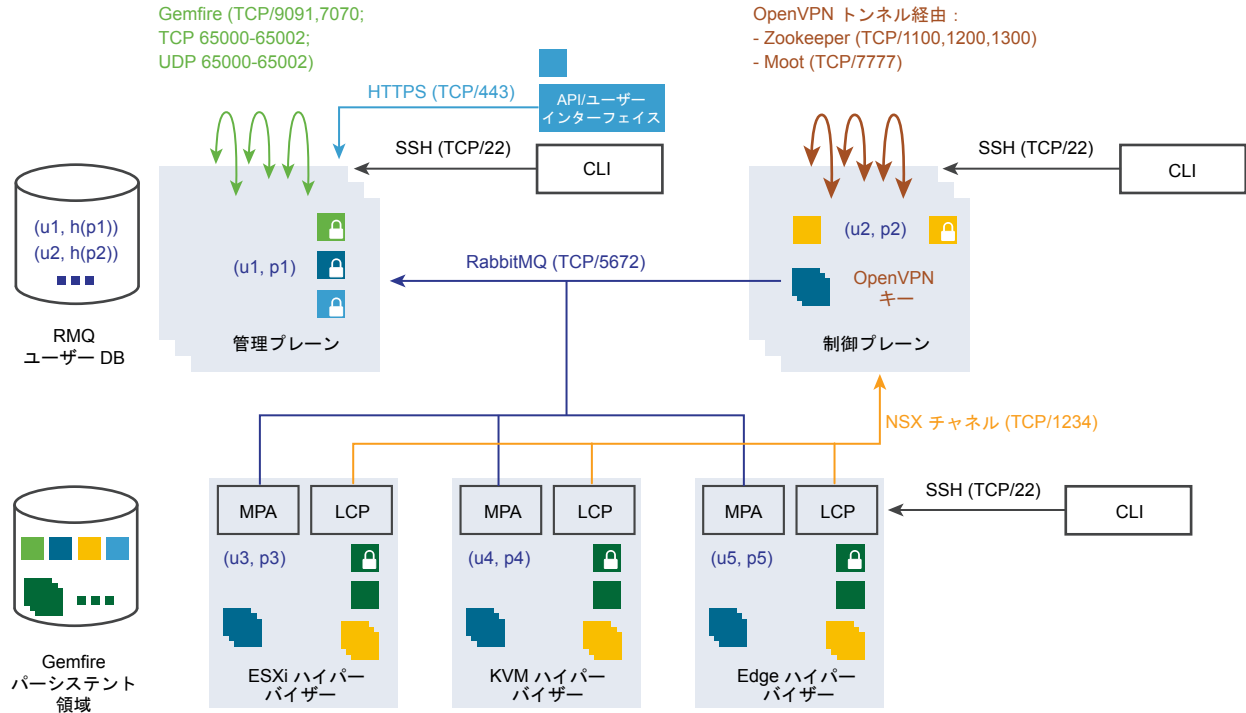
矢印は、通信を開始するエージェントを示しています。デフォルトでは、すべての証明書が自己署名の証明書です。North 方向の API 証明書とプライベート キーは置換できます。

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。

右上に錠のアイコンがある色付きの四角形は、プライベート キーを表します。錠のアイコンがない四角形はパブリック キーです。



- RMQ : ノードごとのアカウント名/共有シークレット、TLS
- NSX チャンネル : ノードごとのクライアント証明書、TLS
- クラスターリング : OpenVPN 経由での Zookeeper と Moot、共有シークレットまたは証明書による認証
- Gemfire : ノードごとのクライアント証明書、TLS
- API/ユーザー インターフェイス : セッション認証、HTTPS
- CLI : ユーザー名/パスワード、SSH

- Gemfire サーバ証明書、プライベート キー
- RMQ サーバ証明書、プライベート キー
- NSX チャンネル サーバ証明書、プライベート キー
- API サーバ証明書、プライベート キー
- (u2, p2) RMQ アカウント名/共有シークレット (ホスト/Edge/CCP ごとに一意)
- NSX チャンネル クライアント証明書、プライベート キー

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

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。これらのポートはファイアウォールでオープンにしておく必要があります。

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 2-6. NSX Manager が使用する TCP および UDP ポート

ソース	ターゲット	ポート	プロトコル	説明
任意	マネージャ	22	TCP	SSH
任意	マネージャ	123	UDP	NTP
任意	マネージャ	443	TCP	NSX API サーバ
任意	マネージャ	161	UDP	SNMP
任意	マネージャ	8080	TCP	インストールとアップグレードの HTTP リポジトリ
任意	マネージャ	5671	TCP	NSX メッセージング
マネージャ	任意	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)
マネージャ	任意	53	TCP	DNS
マネージャ	任意	53	UDP	DNS
マネージャ	任意	123	UDP	NTP
マネージャ	任意	161、162	TCP	SNMP
マネージャ	任意	161、162	UDP	SNMP
マネージャ	任意	514	TCP	Syslog
マネージャ	任意	514	UDP	Syslog
マネージャ	任意	6514	TCP	Syslog
マネージャ	任意	6514	UDP	Syslog
マネージャ	任意	9000	TCP	Log Insight エージェント
マネージャ	任意	33434 - 33523	UDP	Traceroute

NSX Controller によって使用される 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。これらのポートはファイアウォールでオープンにしておく必要があります。

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。指定する場合は、ファイアウォールを適切に設定する必要があります。

表 2-7. NSX Controller によって使用される TCP および UDP ポート

ソース	ターゲット	ポート	プロトコル	説明
任意	コントローラ	22	TCP	SSH
任意	コントローラ	53	UDP	DNS
任意	コントローラ	123	UDP	NTP
任意	コントローラ	161	UDP	SNMP
任意	コントローラ	1100	TCP	Zookeeper クォーラム
任意	コントローラ	1200	TCP	Zookeeper リーダー選出

表 2-7. NSX Controller によって使用される TCP および UDP ポート (続き)

ソース	ターゲット	ポート	プロトコル	説明
任意	コントローラ	1300	TCP	Zookeeper サーバ
任意	コントローラ	1234	TCP	CCP-netcpa 通信
任意	コントローラ	7777	TCP	Moot RPC
任意	コントローラ	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ポートをオープンにする必要があります。
任意	コントローラ	33434 - 33523	UDP	Traceroute
コントローラ	任意	22	TCP	SSH
コントローラ	任意	53	UDP	DNS
コントローラ	任意	53	TCP	DNS
コントローラ	任意	80	TCP	HTTP
コントローラ	任意	123	UDP	NTP
コントローラ	任意	5671	TCP	NSX メッセージング
コントローラ	任意	7777	TCP	Moot RPC
コントローラ	任意	9000	TCP	Log Insight エージェント
コントローラ	任意	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ポートをオープンにする必要があります。
コントローラ	任意	8080	TCP	NSX アップグレード
コントローラ	任意	33434 - 33523	UDP	Traceroute
コントローラ	任意	514	UDP	Syslog
コントローラ	任意	514	TCP	Syslog
コントローラ	任意	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。これらのポートはファイアウォールでオープンにしておく必要があります。

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 2-8. NSX Edge が使用する TCP および UDP ポート

ソース	ターゲット	ポート	プロトコル	説明
任意	Edge	22	TCP	SSH
任意	Edge	123	UDP	NTP
任意	Edge	161	UDP	SNMP
任意	Edge	67、68	UDP	DHCP
任意	Edge	1167	TCP	DHCP バックエンド
任意	Edge	3784、3785	UDP	BFD
任意	Edge	5555	TCP	パブリック クラウド
任意	Edge	6666	TCP	パブリック クラウド
任意	Edge	8080	TCP	NAPI、NSX アップグレード
任意	Edge	2480	TCP	Nestdb
Edge	任意	22	TCP	SSH
Edge	任意	53	UDP	DNS
Edge	任意	80	TCP	HTTP
Edge	任意	123	UDP	NTP
Edge	任意	161、162	UDP	SNMP
Edge	任意	161、162	TCP	SNMP
Edge	任意	179	TCP	BGP
Edge	任意	443	TCP	HTTPS
Edge	任意	514	TCP	Syslog
Edge	任意	514	UDP	Syslog
Edge	任意	1167	TCP	DHCP バックエンド
Edge	任意	1234	TCP	netcpa
Edge	任意	3000 - 9000	TCP	メタデータ プロキシ
Edge	任意	5671	TCP	NSX メッセージング
Edge	任意	6514	TCP	TLS を介した Syslog
Edge	任意	33434 - 33523	UDP	Traceroute

Key Manager が使用する TCP ポート

Key Manager は、特定の TCP 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。これらのポートはファイアウォールでオープンにしておく必要があります。

表 2-9. Key Manager によって使用される TCP ポート

送信元	宛先	ポート	プロトコル	説明
任意	Key Manager	22	TCP	SSH
MP	Key Manager	8992	TCP	管理プレーンと Key Manager の通信
ハイパーバイザー	Key Manager	8443	TCP	ハイパーバイザーと Key Manager の通信
Key Manager	任意	22	TCP	SSH

インストールの概要

一般に、初めてインストールする際の手順は次のとおりです。

- 1 NSX Manager をインストールします。
- 2 NSX Controller をインストールします。
- 3 NSX Controller を管理プレーンに追加します。
- 4 コントロール クラスタを初期化してマスター コントローラを作成します。
環境内に NSX Controller を 1 台だけ配置する場合もこの操作は必要です。
- 5 NSX Controller をコントロール クラスタに追加します。
- 6 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します。
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。
- 7 ハイパーバイザー ホストを管理プレーンに追加します。
これにより、ホストからホスト証明書が管理プレーンに送信されます。
- 8 NSX Edge をインストールします。
- 9 NSX Edge を管理プレーンに追加します。
- 10 トランSPORT ゾーンとトランSPORT ノードを作成します。

これにより、NSX-T のホストスイッチが各ホストに作成されます。このとき、管理プレーンはホストの証明書を制御プレーンに送信し、制御プレーンの情報をホストにプッシュします。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。

以上が推奨される手順ですが、このとおりに行う必要はありません。

NSX Manager はいつでもインストールできます。

NSX Controller は、いつでもインストールして管理プレーンに追加できます。

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 (Fabric)] > [ホスト (Hosts)] > [追加 (Add)] または **POST fabric/nodes** API を使用して両方の処理を同時に行うこともできます。

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、分散論理ルーター、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。

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 を使用する計画があり、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。

手順

- 1 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	
ショ	コマンド
Ubuntu	<pre>apt-get install -y qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils virtinst virt-manager virt-viewer</pre>
RHEL	<pre>yum groupinstall "Virtualization Hypervisor" yum groupinstall "Virtualization Client" yum groupinstall "Virtualization Platform" yum groupinstall "Virtualization Tools"</pre>

- 2 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。

3 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</pre> <pre>INFO: /dev/kvm exists KVM acceleration can be used</pre>
RHEL	<pre>lsmod grep kvm</pre> <pre>kvm_intel 53484 6 kvm 316506 1 kvm_intel</pre>

- 4 (KVM を NSX Manager または NSX Controller のホスト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のために使われています。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	/etc/network/interfaces ファイルを次のように編集します。 <pre data-bbox="443 636 743 873">auto lo iface lo inet loopback auto eth0 iface eth0 inet manual auto br0 iface br0 inet dhcp bridge_ports eth0</pre>
RHEL	/etc/sysconfig/network-scripts/ifcfg-ens32 ファイルを次のように編集します。 <pre data-bbox="443 974 743 1211">DEVICE="ens32" TYPE="Ethernet" NAME="ens32" UUID="<something>" BOOTPROTO="none" HWADDR="<something>" ONBOOT="yes" NM_CONTROLLED="no" BRIDGE="br0"</pre> /etc/sysconfig/network-scripts/ifcfg-br0 ファイルを次のように編集します。 <pre data-bbox="443 1325 727 1457">DEVICE="br0" BOOTPROTO="dhcp" NM_CONTROLLED="no" ONBOOT="yes" TYPE="Bridge"</pre>

- 5 (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のために使われています。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。

前の手順よりもインターフェイスを 1 つ多く設定します。

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	/etc/network/interfaces ファイルを次のように編集します。 <pre data-bbox="443 457 743 779"> 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	/etc/sysconfig/network-scripts/ifcfg-ens32 ファイルを次のように編集します。 <pre data-bbox="443 873 743 1115"> DEVICE="ens32" TYPE="Ethernet" NAME="ens32" UUID="<something>" BOOTPROTO="none" HWADDR="<something>" ONBOOT="yes" NM_CONTROLLED="no" BRIDGE="br0" </pre> /etcsysconfig/network-scripts/ifcfg-ens33 ファイルを次のように編集します。 <pre data-bbox="443 1230 743 1440"> DEVICE="ens33" TYPE="Ethernet" NAME="ens33" UUID="<something>" BOOTPROTO="none" HWADDR="<something>" ONBOOT="yes" NM_CONTROLLED="no" </pre> /etcsysconfig/network-scripts/ifcfg-br0 ファイルを次のように編集します。 <pre data-bbox="443 1556 727 1692"> 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 などの個別のネットワーク設定ファイルは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。

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」が自動的に作成されます。Ubuntu では、/etc/network/interfaces に次のようなエントリが記載されます。

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

RHEL では、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載されます。

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

- 6 ネットワークの変更を有効にするには、ネットワークを再起動するか、Linux サーバを再起動します。
- 7 コア ダンプ用にホストを準備します。

Linux ディストリビューション	
ビューション	コア ダンプの準備
RHEL	<p>次のコマンドを実行します。</p> <pre>mkdir /var/cores chmod 1777 /var/cores echo "kernel.core_pattern = /var/cores/core.%e.%t.%p" >> /etc/sysctl.conf sysctl -p</pre> <p>/etc/security/limits.conf に次の行を追加します。</p> <pre>* soft core unlimited * hard core unlimited root soft core unlimited root hard core unlimited</pre>

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 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、分散論理ルーター、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 は vSphere ESXi または KVM でサポートされています。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。vSphere の高可用性 (HA) 機能を使用して NSX Manager の可用性を維持することができます。ESXi では、NSX Manager アプライアンスを共有ストレージにインストールすることを推奨します。vSphere 高可用性では共有ストレージが必要です。それによって、元のホストに障害が発生したときに NSX Manager アプライアンスを別のホストで再起動できます。

NSX Manager は、次の展開方法をサポートしています。

- OVA/OVF
- QCOW2

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 アプライアンスでは、パスワードの複雑性に対して次の要件が定められています。

- 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こと
- パリンドローム (回文) になっていないこと

パスワードが複雑性の要件を満たしていなくてもインストールは正常に終了します。展開時にユーザー **admin** に指定したパスワードが十分に複雑でない場合は、展開後に **admin** でログインし、パスワード変更のプロンプトに沿って変更する必要があります。ユーザー **root** のパスワードが十分に複雑でない場合も、**admin** でログインして次のコマンドでパスワードを変更します。

```
set user root password <password>
```

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あとは、NSX Manager の起動に数分かかる場合があります。

アプライアンス上のコア サービスは、十分に複雑なパスワードが設定されるまで開始されません。

NSX Manager を OVA ファイルからデプロイした後は、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 設定を変更することはできません。

NSX Manager のインストール時には、アンダースコアを含まないホスト名を選択します。アンダースコアを含むホスト名を指定すると、展開後にアプライアンスのデフォルトのホスト名が nsx-manager のようになります。

重要: NSX 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 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。

この章には、次のトピックが含まれています。

- [vSphere Web Client を使用した ESXi への NSX Manager 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](#)

vSphere Web Client を使用した ESXi への NSX Manager のインストール

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

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 がいない場合は、**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。

前提条件

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

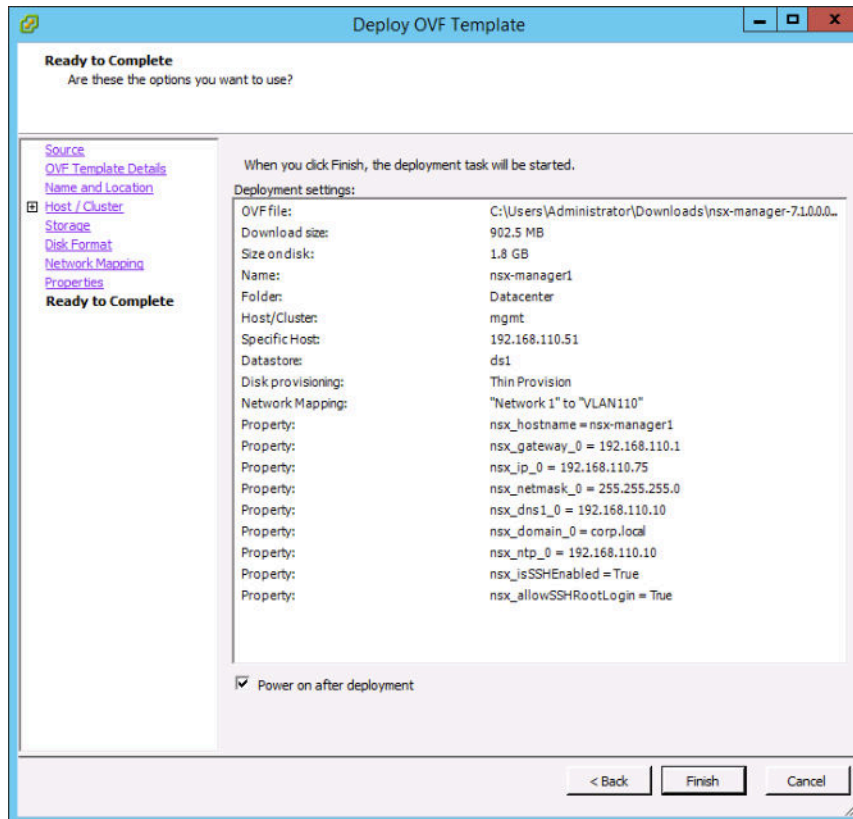
- IPv4 IP アドレス スキームを計画します。NSX-T のこのリリースでは、IPv6 はサポートされていません。

手順

- 1 NSX Manager 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 (Deploy OVF template)] 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。
たとえば、vSphere Distributed Switch を使用している場合は、NSX Manager を Mgmt_VDS - Mgmt というポート グループに配置できます。

7 NSX Manager のパスワードと IP アドレス を設定します。

すべてのオプションを設定した後の最終的な確認画面の一例を示します。



8 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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 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 コンポーネントが完全に起動した後、管理者として 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 コンポーネントに必要な接続があることを確認します。

- NSX コンポーネントに ping を送信できることを確認します。

- NSX コンポーネントがそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX コンポーネントが、NSX コンポーネントと同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX コンポーネントがその DNS サーバと NTP サーバに ping を送信でき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 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、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。

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 or hostname of NSX Manager>` です。たとえば、`https://192.168.110.75` のようになります。

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場合は、コマンドライン ユーティリティである VMware OVF Tool を使用します。

デフォルトでは、`nsx_isSSHEntabled` 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_isSSHEntabled` 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

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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。次に例を示します。

```
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_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_isSSHEntabled=<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
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
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。次に例を示します。

```

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_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
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully

```

- 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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 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 コンポーネントが完全に起動した後、管理者として 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 コンポーネントに必要な接続があることを確認します。

- NSX コンポーネントに ping を送信できることを確認します。

- NSX コンポーネントがそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX コンポーネントが、NSX コンポーネントと同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX コンポーネントがその DNS サーバと NTP サーバに ping を送信でき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 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、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。

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 or hostname of NSX Manager>` です。たとえば、`https://192.168.110.75` のようになります。

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。
- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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

- 1 NSX Manager の QCOW2 イメージをダウンロードし、適切な場所にコピーします。
- 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_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"/>
    <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。

複数のマネージャを作成する場合は、マネージャごとに QCOW2 イメージのコピーを別に作成します。**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。

```
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-1.1.0.0.0.4446302.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 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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 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 コンポーネントの起動が完了したら、3 分待ってから管理者として CLI にログインします。エンド ユーザー使用許諾契約書 (EULA) の画面が表示されます。EULA に同意します。その後、**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-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 コンポーネントに必要な接続があることを確認します。

- NSX コンポーネントに ping を送信できることを確認します。
- NSX コンポーネントがそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX コンポーネントが、NSX コンポーネントと同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX コンポーネントがその DNS サーバと NTP サーバに ping を送信でき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 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、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。

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 or hostname of NSX Manager>` です。たとえば、`https://192.168.110.75` のようになります。

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。

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。これは、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するもので、すべてのホスト、論理スイッチ、および分散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-T のホストスイッチや Open vSwitch (OVS) などの仮想スイッチは、ESX や、KVM などその他のハイパーバイザーの中に存在します。

NSX Controller では、次の展開方法がサポートされています。

- OVA/OVF
- QCOW2

NSX Controller は ESX と KVM でサポートされています。

PXE ブートによる NSX Controller のインストールはサポートされていません。

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。

NSX-T アプライアンスでは、パスワードの複雑性に対して次の要件が定められています。

- 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こと
- パリンドローム（回文）になっていないこと

パスワードが複雑性の要件を満たしていなくてもインストールは正常に終了します。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。

注: 複雑さの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上で起動されません。

NSX Controller を OVA ファイルからデプロイした後は、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 設定を変更することはできません。

NSX Manager のインストール時には、アンダースコアを含まないホスト名を選択します。そうでない場合は、ホスト名が localhost に設定されます。

重要: NSX 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 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。

この章には、次のトピックが含まれています。

- [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 Controller の管理プレーンへの追加](#)
- [コントロール クラスターの初期化によるコントロール クラスター マスターの作成](#)
- [クラスター マスターを使用した NSX Controller の追加](#)

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には、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、たとえば vCenter Server に接続している vSphere Client を使用できます。

バックアップとリストアをサポートするには、NSX Controller アプライアンスに固定管理 IP アドレスがあることが必要です。DHCP を使用して管理 IP アドレスを割り当てることはできません。管理 IP アドレスの変更はサポートされません。バックアップとリストアの情報については、『NSX-T 管理ガイド』を参照してください。

パスワード強度の基準に準拠したパスワードを使用する必要があります。NSX-T アプライアンスでは、複雑性に関する次の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こと
- パリンドローム（回文）になっていないこと

PXE のインストールでは、root ユーザーと管理者ユーザーのパスワードに、sha-512 アルゴリズムで暗号化したパスワード文字列を指定する必要があります。

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: 複雑さの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上で起動されません。

重要: NSX 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 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

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。
- IPv4 IP アドレス スキームを計画します。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 ホスト上に OVF テンプレートを展開する権限が必要です。
- アンダースコア (_) が含まれていないホスト名を選択します。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)。

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 (Deploy OVF template)] 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。

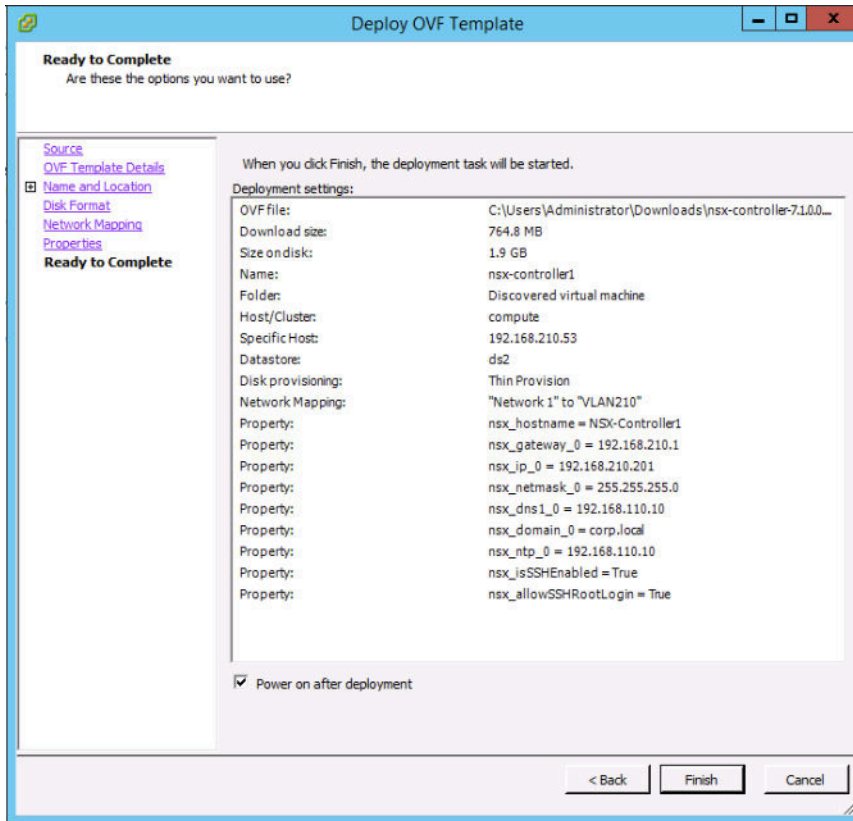
通常は、ネットワーク管理機能を備えたクラスタに NSX Controller を配置します。

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。

たとえば、vSphere Distributed Switch を使用している場合は、NSX Controller を Mgmt_VDS - Mgmt というポート グループに配置できます。

7 NSX Controller のパスワードと IP アドレスの設定を行います。

すべてのオプションを設定した後の最終的な確認画面の一例を示します。



8 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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 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 コンポーネントが完全に起動した後、管理者として 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 コンポーネントに必要な接続があることを確認します。

- NSX コンポーネントに ping を送信できることを確認します。
- NSX コンポーネントがそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX コンポーネントが、NSX コンポーネントと同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX コンポーネントがその DNS サーバと NTP サーバに ping を送信でき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 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、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。

次のステップ

NSX Controller を管理プレーンに追加します。[「NSX Controller 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 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 ユーティリティである 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せんが、`root` でログインすることはできません。

前提条件

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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"
--acceptAllEulas
--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_hostname=nsx-controller
<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-controller
Task Completed
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
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-controller
--X:injectOvfEnv
--X:logFile=ovftool.log
```

```

--allowExtraConfig
--datastore=ds1
--network="management"
--acceptAllEulas
--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_hostname=nsx-controller
<path/url to 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%40vsphere.local@192.168.110.24:443/
Deploying to VI: vi://administrator%40vsphere.local@192.168.110.24:443/
Transfer Completed
Powering on VM: nsx-controller
Task Completed
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully

```

- 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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 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 コンポーネントが完全に起動した後、管理者として 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 コンポーネントに必要な接続があることを確認します。

- NSX コンポーネントに ping を送信できることを確認します。
- NSX コンポーネントがそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX コンポーネントが、NSX コンポーネントと同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX コンポーネントがその DNS サーバと NTP サーバに ping を送信でき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 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、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。

次のステップ

NSX Controller を管理プレーンに追加します。[「NSX Controller の管理プレーンへの追加」](#)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、および分散分散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。
- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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

- 1 NSX Controller の 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_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。

複数のコントローラを作成する場合は、コントローラごとに QCOW2 イメージのコピーを別に作成します。**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。

```
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 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-Controller1
Escape character is ^]

nsx-Controller1 login:

```

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

6 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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 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 コンポーネントが完全に起動した後、管理者として 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 コンポーネントに必要な接続があることを確認します。

- NSX コンポーネントに ping を送信できることを確認します。
- NSX コンポーネントがそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX コンポーネントが、NSX コンポーネントと同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX コンポーネントがその DNS サーバと NTP サーバに ping を送信でき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 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、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。

次のステップ

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

NSX Controller の管理プレーンへの追加

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

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。

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。次に例を示します。

```
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。

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 アドレスと任意でポート番号
- 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-Controller1> join management-plane NSX-Manager username admin thumbprint <NSX-Manager's-thumbprint>
Password for API user: <NSX-Manager's-password>
Node successfully registered and controller restarted
```

このコマンドを各コントローラ ノードで実行します。

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
```

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 クラスタを初期化しなかった場合、コントローラがハイパーバイザー ホストと通信できません。

前提条件

- NSX Controller を少なくとも 1 つインストールします。
- NSX Controller を管理プレーンに追加します。
- 共有シークレット パスワードを選択します。共有シークレット パスワードは、ユーザー定義の共有シークレット パスワード（たとえば「secret123」）です。パスワードは、クラスタ内の 3 台のノードで共通である必要があります。

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` コマンドを実行し、プロンプトが表示されたら共有シークレットを入力します。
- 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 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.
```

`get control-cluster status verbose` コマンドを実行して、`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 id	vpn address	rpc address status	rpc port	global
557a911f-41fd-4977-9c58-f3ef55b3efe7	10.0.0.1	192.168.110.34 connected	7777	1

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]
(queueued=0,recved=60324,sent=60324,sid=0x100000f14a10003,lop=PING,est=1459376913497,to=30000,
lxcid=0x8,lzxid=0x10000017a,lresp=604617273,llat=0,minlat=0,avglat=0,maxlat=1088)
/10.0.0.1:35462[0](queueued=0,recved=1,sent=0)
/10.0.0.1:51724[1]
(queueued=0,recved=45786,sent=45803,sid=0x100000f14a10001,lop=GETC,est=1459376911226,to=40000,
lxcid=0x21e,lzxid=0x10000017a,lresp=604620658,llat=0,minlat=0,avglat=0,maxlat=1841)
/10.0.0.1:51725[1]
(queueued=0,recved=60328,sent=60333,sid=0x100000f14a10002,lop=PING,est=1459376913455,to=30000,
lxcid=0xc,lzxid=0x10000017a,lresp=604618294,llat=0,minlat=0,avglat=0,maxlat=1356)
/10.0.0.1:51730[1]
(queueued=0,recved=45315,sent=45324,sid=0x100000f14a10006,lop=PING,est=1459376914516,to=40000,
lxcid=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、少なくとも 1 つの NSX Controller が常に使用可能になります。

前提条件

- 3 台の NSX Controller アプライアンスをインストールします。
- 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 Controller の管理プレーンへの追加」](#)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。
- **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 コンポーネント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、共有シークレット 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 で入力する共有シークレット パスワードは、NSX-Controller1 で入力した共有シークレット 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、任意でマスター以外（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、複数のコントローラ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。

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。各コントローラ ノードにも UUID があります。

注: 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コントローラで **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

ハイパーバイザー ホストを NSX-T のファブリックに追加します。[章 7 「ホストの準備」](#) 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) を行う 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。

NSX Edge では、次の展開方法がサポートされています。

- OVA/OVF
- PXE を使用した ISO
- PXE を使用しない ISO

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 のインストールでは、root ユーザーと管理者ユーザーのパスワードに、sha-512 アルゴリズムで暗号化したパスワード文字列を指定する必要があります。

NSX-T アプライアンスでは、パスワードの複雑性に対して次の要件が定められています。

- 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こと
- パリンドローム（回文）になっていないこと

パスワードが複雑性の要件を満たしていなくてもインストールは正常に終了します。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。

注: 複雑さの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上で起動されません。

NSX Edge を OVA ファイルからデプロイした後は、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 設定を変更することはできません。

NSX Manager のインストール時には、アンダースコアを含まないホスト名を選択します。アンダースコアを含むホスト名を指定すると、展開後にアプライアンスのデフォルトのホスト名が `nsx-manager` のようになります。

重要: NSX 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 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。

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](#)
- [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](#)

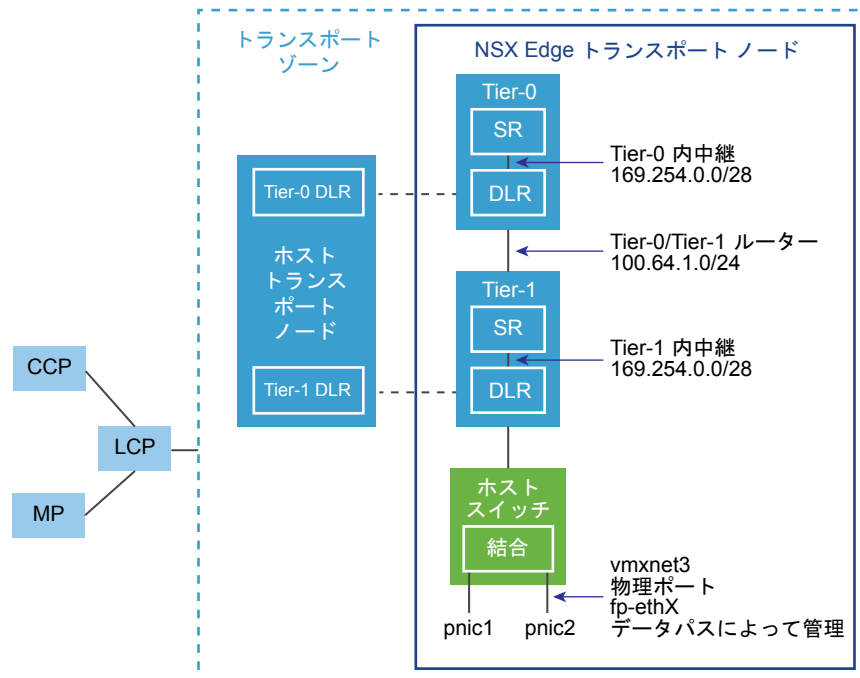
NSX Edge のネットワーク設定

NSX Edge は、ISO、OVA/OVF、PXE ブートを使用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 つのトランスポート ノードがあります。1 つのトランスポート ノードはホストです。もう 1 つは NSX Edge です。

図 6-1. NSX Edge の概要図



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。分散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 の分散論理ルーターの処理を支えます。各分散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 DR 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分散論理ルーターを設定する場合に必要です。Tier-0 の分散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、SR と DR 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 の分散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 の分散論理ルーターでは、この分散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。MP と CCP 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。

最後に、図には、高可用性のために結合された 2 つの物理 NIC (pnic1 と pnic2) の例を示しています。これらの物理 NIC はデータパスによって管理されます。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。

各 NSX Edge には少なくとも 2 つの物理リンクを割り当てるのが推奨されます。任意で、同じ物理 NIC 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、リンク vnic1 が最初に見つかる場合があります。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。

物理リンク情報は NSX Edge の CLI で **get interfaces** と **get physical-ports** の各コマンドを実行して確認できます。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 ゾーンとホストスイッチ

NSX Edge のネットワークを理解するには、トランスポート ゾーンとホストスイッチ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。ホストスイッチは、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。ホストスイッチの目的は、分散論理ルーターのアップリンクとダウンリンクを物理 NIC にバインドすることです。NSX Edge が属するトランスポート ゾーンごとに、NSX Edge に単一のホストスイッチ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 Edge は 1 つのオーバーレイ トランスポート ゾーンにのみ属することができます。
- NSX-T 外部のアップリンク用 VLAN : NSX Edge が属することができる VLAN トランスポート ゾーンの数に制限はありません。

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 ゾーンがゼロ個の場合も、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ホストスイッチを使用できるからです。各 NSX Edge にホストスイッチを 1 つだけ設定する場合は、このようにします。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 トラフィックとその他の VLAN トラフィックに同じ VLAN ID を使用する必要がある場合は、これらを 2 つの異なるホストスイッチで設定する必要があります。1 つは VLAN 用、もう 1 つはオーバーレイ用です。

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

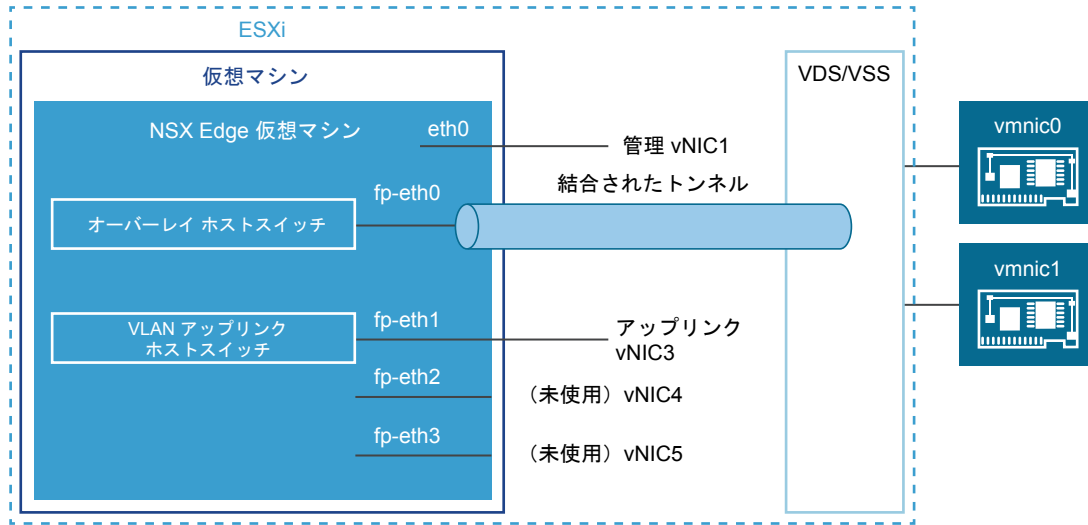
NSX Edge を仮想アプライアンスまたは仮想マシンとしてインストールすると、fp-ethX という内部インターフェイスが作成されます。ここで X は 0、1、2、3 です。これらのインターフェイスは、トップオブブラック (ToR) スイッチへのアップリンク用と、NSX-T のオーバーレイ トンネル用に割り当てられます。

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 に少なくとも 2 つの vmnics を割り当てます。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。

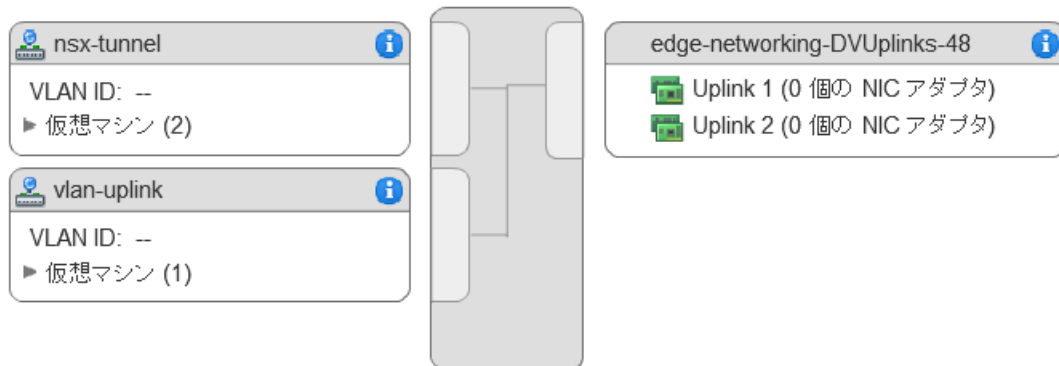
次の物理トポロジ例では、fp-eth0 を NSX-T のオーバーレイ トンネルに使用しています。fp-eth1 は VLAN アップリンクに使用しています。fp-eth2 と fp-eth3 は使用されていません。

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 つのホストスイッチがあります。

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 を使用する場合、この例の仮想マシン ポート グループと一致するためには、ネットワークの ovftool 設定を次のようにします。

```
--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。これは一例です。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 (Host)] > [設定 (Configuration)] > [ネットワーク (Networking)] > [ネットワークの追加 (Add Networking)] > [仮想マシン (Virtual Machine)] > [VLAN ID すべて (4095) (VLAN ID All (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。

NSX Edge とホスト トランスポート ノードは同じハイパーバイザーに展開できます。

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動し、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。

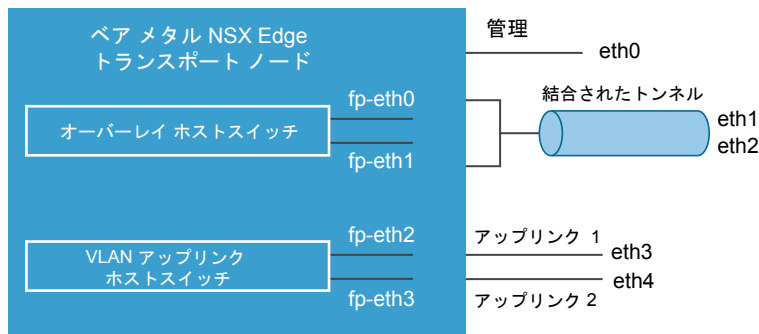
ベア メタルの NSX Edge ネットワーク

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 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すべてまたは一部を、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. ベア メタルの NSX Edge ネットワークのリンク設定の一例



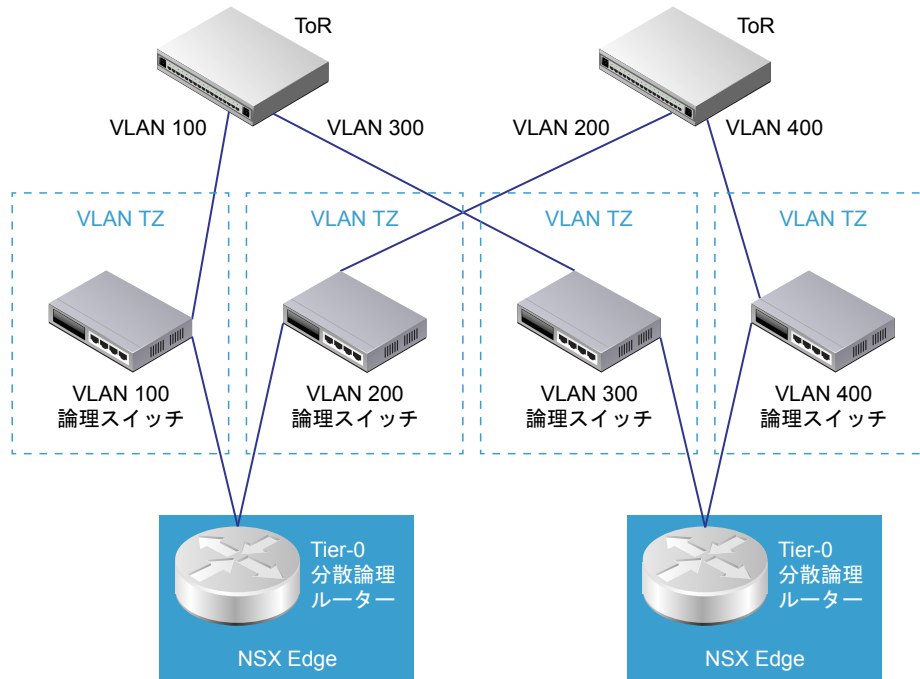
NSX Edge のアップリンクの冗長性

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のために 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、新しいホストスイッチ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 つのホストスイッチ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



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には、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、たとえば vCenter Server に接続している vSphere Client を使用できます。

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

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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 ホスト上に OVF テンプレートを展開する権限が必要です。

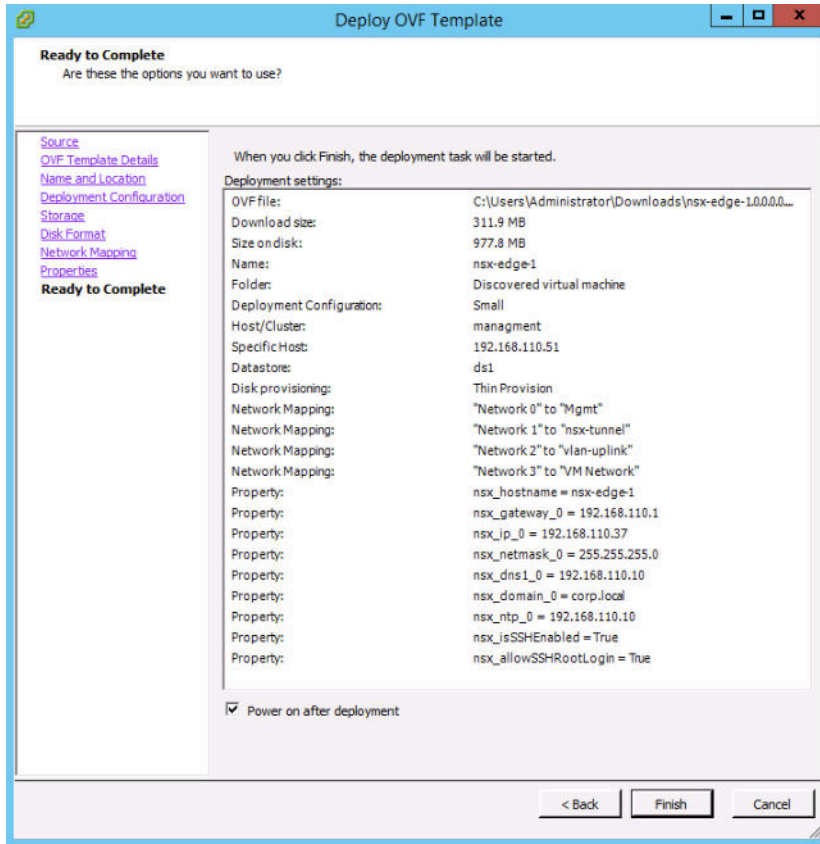
- アンダースコア (_) が含まれていないホスト名を選択します。そうでない場合は、ホスト名が <localhost> に設定されます。
- OVF テンプレートを展開できる管理ツールが必要です (vCenter Server や vSphere Client など)。
手動で設定できるようにするには、OVF 展開ツールで設定オプションがサポートされている必要があります。
- クライアント統合プラグインがインストールされている必要があります。
- NSX Edge のネットワーク要件を、[「NSX Edge のネットワーク設定」](#) で確認してください。

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 (Deploy OVF template)] 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、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 Edge に権限が適用されます。
- 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 アドレスの設定を行います。

すべてのオプションを設定した後の最終的な確認画面の一例を示します。



9 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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。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。

NSX Edge が完全に起動した後、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-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 アプライアンスに必要な接続がされていることを確認します。

- NSX Edge に ping を送信できることを確認します。
- NSX Edge がそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX Edge が、NSX Edge と同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX Edge が、自身の DNS サーバと NTP サーバに ping を送信でき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。

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 が誤った NIC を管理 NIC として割り当てた場合は、次のように修正できます。

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 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。
- 4 `start service dataplane`

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 ユーティリティである 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

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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 のネットワーク要件を、[「NSX Edge のネットワーク設定」](#)で確認してください。
- ESXi ホスト上に OVF テンプレートを展開する権限が必要です。
- アンダースコア (`_`) が含まれていないホスト名を選択します。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
Warning:
  - No manifest entry found for: 'nsx-<component>.ovf'.
  - File is missing from the manifest: 'nsx-<component>.ovf'.
  - ExtraConfig options exists in source.
  - Skipping monitor as the --X:waitForIp option is not given.
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

```

Warning:

- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.

Completed successfully

- 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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。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。

NSX Edge が完全に起動した後、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-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 アプライアンスに必要な接続がされていることを確認します。

- NSX Edge に ping を送信できることを確認します。
- NSX Edge がそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX Edge が、NSX Edge と同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX Edge が、自身の DNS サーバと NTP サーバに ping を送信でき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。

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 が誤った NIC を管理 NIC として割り当てた場合は、次のように修正できません。

```
1 stop service dataplane
```

```
2 set interface eth0 dhcp plane mgmt
```

3 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。

```
4 start service dataplane
```

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。

この手順では、Ubuntu で PXE サーバを設定する方法を示します。PXE は DHCP と TFTP の 2 つ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 サービスを求めるネットワーク 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を提供します。

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を示します。

前提条件

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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 のネットワーク要件を、[「NSX Edge のネットワーク設定」](#) で確認してください。
- 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。

手順

- 1 (オプション) kickstart ファイルを作成します。

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

kickstart ファイルは、

```
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 ファイルには、必要な 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
```

管理者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 <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 DHCP の設定を定義します。

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 サービスが実行されていることを確認します。

```
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 インストーラの ISO ファイルを適切な場所にコピーまたは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
```

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...
```

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。

`usermod` コマンドを `preseed.cfg` に追加して、root、管理者、またはその両方のパスワードを設定することもできます。たとえば、次の 2 行を追加できます。

```
usermod --password '$6$VS3exId0aKmw\U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/'
root; \
usermod --password '$6$VS3exId0aKmw\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 6tgml...` で設定したパスワードに取って代わります。

`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=trusty --
```

- 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: エラーが返された場合（例：「stop: Unknown instance: start: Job failed to start」）、`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、エラーの発生元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 仮想マシン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。

ネットワークが自動的に設定され、パーティションが作成され、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、管理者または 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、管理者のログイン パスワードは [default] です。

- 23 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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。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。

NSX Edge が完全に起動した後、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-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 アプライアンスに必要な接続がされていることを確認します。

- NSX Edge に ping を送信できることを確認します。
- NSX Edge がそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX Edge が、NSX Edge と同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX Edge が、自身の DNS サーバと NTP サーバに ping を送信でき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。

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 が誤った NIC を管理 NIC として割り当てた場合は、次のように修正できます。

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 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。
- 4 `start service dataplane`

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。このインストール方法は通常、PXE サーバにアクセスできない事前検証 (POC) 環境で使用します。

前提条件

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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 のネットワーク要件を、[「NSX Edge のネットワーク設定」](#)で確認してください。

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 このディスクからホストを起動します。
- 3 [自動インストール (Automated installation)] を選択します。

Enter キーを押した後、10 秒程度の間があ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 アドレス設定を求められます。

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

- 4 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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。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。

NSX Edge が完全に起動した後、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-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 アプライアンスに必要な接続がされていることを確認します。

- NSX Edge に ping を送信できることを確認します。
- NSX Edge がそのデフォルト ゲートウェイに ping を送信できることを確認します。
- NSX Edge が、NSX Edge と同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX Edge が、自身の DNS サーバと NTP サーバに ping を送信でき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。

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 が誤った NIC を管理 NIC として割り当てた場合は、次のように修正できます。

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 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。
- 4 `start service dataplane`

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 デバイスは、ISO ファイルを使用して手動でインストールできます。このインストール方法は通常、PXE サーバにアクセスできない事前検証 (POC) 環境で使用します。

重要: NSX 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 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

- システム要件が満たされ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、ターゲット仮想マシン ポート グループ ネットワークを作成します。ほとんどの展開では、NSX アプライアンスが管理仮想マシン ネットワークに置かれます。

複数の管理ネットワークが存在する場合は、NSX 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 アプライアンスが通信を行う管理仮想マシン ポート グループを準備します。

- IPv4 IP アドレス スキームを計画します。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 のネットワーク要件を、[「NSX Edge のネットワーク設定」](#) で確認してください。

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 (Connect at power on)] 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 (Automated installation)] を選択します。
Enter キーを押した後、10 秒程度の間がある可能性があります。

パワーオン中に、仮想マシンから 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 アドレス設定を求められます。

デフォルトでは、root のログイン パスワードは [vmware] で、管理者のログイン 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こと
- パリンドローム（回文）になっていないこと

重要: 複雑さの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上で起動されません。

4 最適なパフォーマンスを実現するように、NSX 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保証された下限であり、メモリがオーバーコミットされる場合でも、この容量が保証されます。NSX 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。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。

NSX Edge が完全に起動した後、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-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 アプライアンスに必要な接続がされていることを確認します。

- NSX Edge に ping を送信できることを確認します。
- NSX Edge がそのデフォルト ゲートウェイに ping を送信できることを確認します。

- NSX Edge が、NSX Edge と同じネットワークに置かれたハイパーバイザー ホストに ping を送信できることを確認します。
- NSX Edge が、自身の DNS サーバと NTP サーバに ping を送信でき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。

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 が誤った NIC を管理 NIC として割り当てた場合は、次のように修正できます。

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 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。
- 4 `start service dataplane`

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。

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 に固有の一連の数値です。

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 アドレスと任意でポート番号
- 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 (Fabric)] > [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](#)
- [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。

手順

- Ubuntu 14.04 の場合は次のコマンドを実行します。

```
apt-get install libunwind8 libgflags2 libgoogle-perftools4 traceroute
apt-get install python-mako python-simplejson python-support python-unittest2 python-
yaml python-netaddr
apt-get install libprotobuf8
apt-get install libboost-filesystem1.54.0 libboost-chrono1.54.0
apt-get install dkms
```

- Ubuntu 16.04 の場合は次のコマンドを実行します。

```
apt-get install libunwind8 libgflags2v5 libgoogle-perftools4 traceroute
apt-get install python-mako python-simplejson python-unittest2 python-yaml python-netaddr
apt-get install libprotobuf9v5
apt-get install libboost-filesystem1.58.0 libboost-chrono1.58.0 libgoogle-glog0v5
apt-get install dkms
```

- RHEL 7.2 の場合は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"
yum install boost-filesystem PyYAML boost-iostreams boost-chrono python-mako python-
netaddr python-six gperftools-libs libunwind yum-utils wget net-tools redhat-lsb-core
tcpdump wget
```

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。

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
 - (KVM) SHA-256 SSL サムプリント
 - (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。
 - 自分で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
```

- 別の方法として、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
```

- 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、次のコマンドを実行します。

```
# ssh-keyscan -t rsa <hostname> > <hostname>.pub  
# awk '{print $3}' hostname.pub | base64 -d | sha256sum -b | sed 's/ .*$//' | xxd -r  
-p | base64
```

where <hostname> is the hypervisor's hostname or IP address.

- 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-mgr>>) にログインします。
- 3 [ファブリック (Fabric)] - [ノード (Nodes)] - [ホスト (Hosts)] の順に選択し、[追加 (Add)] 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 (Fabric)] > [ノード (Nodes)] > [ホスト (Hosts)] で [展開ステータス：インストール成功 (Deployment Status: Installation Successful)] および [MPA 接続：アップ (MPA Connectivity: Up)] 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 (LCP Connectivity)] 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" : "f36d5a79-233c-47c9-9c17-9adc9f8ec466",
  "display_name" : "10.143.1.177",
  "fqdn" : "w1-mvpcloud-177.eng.vmware.com",
  "ip_addresses" : [ "10.143.1.177" ],
  "external_id" : "f36d5a79-233c-47c9-9c17-9adc9f8ec466",
  "discovered_ip_addresses" : [ "192.168.150.104", "10.143.1.177" ],
  "os_type" : "ESXI",
  "os_version" : "6.5.0",
  "managed_by_server" : "",
  "_create_time" : 1480369243245,
  "_last_modified_user" : "admin",
```



```

    "_last_modified_time" : 1480369243245,
    "_create_user" : "admin",
    "_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" : "1.1.0.0.4649755",
"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 (Fabric)] > [ノード (Nodes)] > [ホスト (Hosts)] > [追加 (Add)] 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の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、Distributed Firewall、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。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 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` : ホストスイッチ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 に追加するには、NSX-T カーネル モジュールを Ubuntu KVM 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の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、Distributed Firewall、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。

- 2 (オプション) /tmp ディレクトリに移動します。

```
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 パッケージを解凍します。

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

- 5 パッケージディレクトリに移動します。

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

- 6 パッケージをインストールします。

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

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	<release>	amd64	NSX Host Status Reporter
	for Aggregation Service			
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	<release>	amd64	NSX Transport Node Status
	Reporter			
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 に追加するには、NSX-T カーネル モジュールを RHEL KVM 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の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、Distributed Firewall、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 -xvf nsx-lcp-<release>-rhel71_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-rhel71_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 マシンが RHEL リポジトリにアクセスできるとの想定のもと、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
```

確認するには、[rpm -qa | grep nsx] コマンドを実行します。

```
user@host:~$ rpm -qa | grep nsx

nsxa-<release>.el7.x86_64.rpm
nsx-agent-<release>.el7.x86_64.rpm
nsx-aggservice-<release>.el7.x86_64.rpm
nsx-cli-<release>.x86_64.rpm
nsx-da-<release>.el7.x86_64.rpm
nsx-host-<release>.x86_64.rpm
```

```

nsx-host_node_status_reporter-<release>.el7.x86_64.rpm
nsx-lldp-<release>.el7.x86_64.rpm
nsx-logical_exporter-<release>.el7.x86_64.rpm
nsx-mpa-<release>.el7.x86_64.rpm
nsx-netcpa-<release>.el7.x86_64.rpm
nsx-sfhc-<release>.el7.x86_64.rpm
nsx-transport_node_status-<release>.el7.x86_64.rpm

```

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 ハイパーバイザー ホストへの 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 に固有の一連の数値です。

次はその例です。

```

NSX-Manager1> get certificate api thumbprint
...

```

- 4 ハイパーバイザー ホストで [/opt/vmware/nsx-cli/bin/scripts/nsxcli] コマンドを実行して NSX-T の 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。

- 5 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 アドレスと任意でポート番号
- 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 (Fabric)] > [ノード (Node)] > [ホスト (Hosts)] で、ホストの MPA 接続が [稼動中 (Up)] 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、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 ESXi ホストの `/etc/vmware/nsx/controller-info.xml` に NSX Controller のアドレスがあるはずです。

```
[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

8

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
- 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
- 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、分散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。

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。

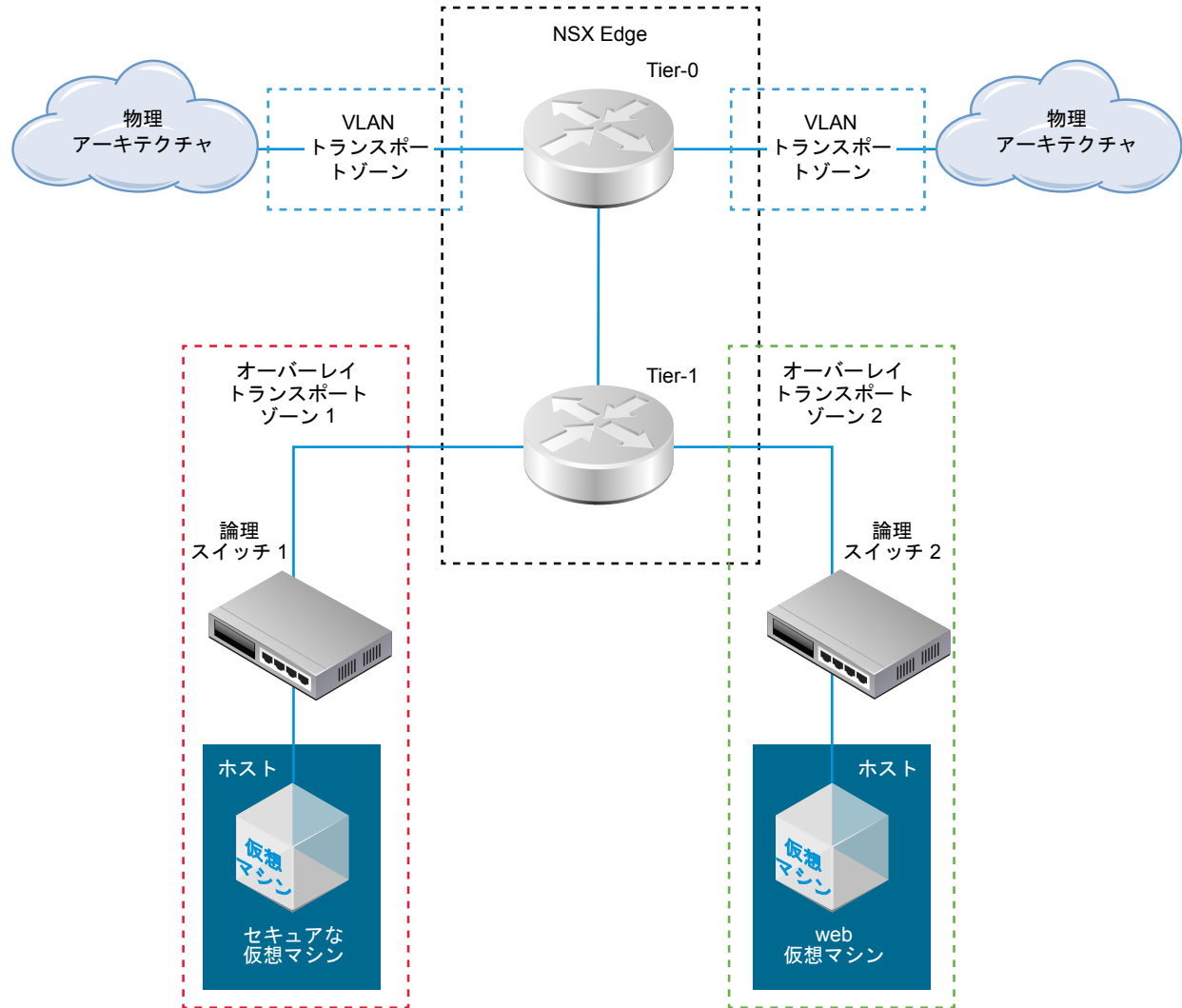
トランスポート ノードは、ハイパーバイザー ホストまたは NSX Edge です。NSX Edge は、複数のトランスポート ゾーンに属することができます。ハイパーバイザー ホストおよび NSX-T 論理スイッチは、1 つのトランスポート ゾーンにのみ属することができます。

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。

NSX Edge トランスポート ノードは複数のトランスポート ゾーンに属することができます (1 つのオーバーレイ トランスポート ゾーンと、複数の VLAN トランスポート ゾーン)。VLAN トランスポート ゾーンは、外部への VLAN アップリンク用です。

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



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 ヘッダで使用する、ソース IP アドレスとターゲット 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 プールを使用できます。

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。

```
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-mgr>>) にログインします。
- 2 [インベントリ (Inventory)] > [IP アドレス プール (IP Pools)] の順に選択し、[追加 (Add)] 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 プロファイルはホストスイッチ プロファイルです。これはハイパーバイザー ホストから NSX-T 論理スイッチへのリンク、または NSX Edge ノードからトップオブラック (ToR) 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。

NSX Edge がベア メタルにインストールされている場合は、デフォルトのアップリンク プロファイルを使用できます。デフォルトのアップリンク プロファイルには、1 つのアクティブ アップリンクと 1 つのパッシブ スタンバイ アップリンクが必要です。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。NSX Edge を仮想アプライアンスとして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せずに、カスタム アップリンク プロファイルを作成する必要があります。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: それぞれ異なる VLAN を使用してアップリンクごとに別のホストスイッチを作成すると、NSX Edge 仮想マシンで複数のアップリンクを設定できます。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 のネットワーク機能について理解している必要があります。[「NSX Edge のネットワーク設定」](#)を参照してください。

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 は現在、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 は現在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。したがって、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-mgr>>) にログインします。
- 2 [ファブリック (Fabric)] > [プロファイル (Profiles)] > [アップリンク プロファイル (Uplink Profiles)] の順に選択し、[追加 (Add)] をクリックします。
- 3 次の情報を入力します。
 - アップリンク プロファイル名
 - (オプション) 説明
 - チーミング ポリシー：フェイルオーバー順序またはロード バランシングのソース（デフォルトはフェイルオーバー順序）
 - フェイルオーバー順序：アクティブ アダプタのリストから、フェイルオーバー検出基準を満たした最上位のアップリンクを常に使用します。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 バランシングのソース：ソースのイーサネット MAC アドレスのハッシュに基づいてアップリンクを選択します。
 - (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)
 - アクティブ アップリンク名のコンマ区切りのリスト
 - (オプション) スタンバイ アップリンク名のコンマ区切りのリスト

ここで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。
 - (オプション) トランスポート VLAN

- MTU（デフォルトは 1600）

次はその例です。

New Uplink Profile

Name: *

Description:

Teaming Policy: *

LAGs

+ INSERT ROW ☐ COLUMNS

Name *	LACP Mode	LACP Load Balancing *	Uplinks	LACP Time Out

Active Uplinks: *

Standby Uplinks:

Transport VLAN:

MTU: *

アップリンク プロファイルは、GET /api/v1/host-switch-profiles API 呼び出しで確認できます。

```
{
  "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": [],
        "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"
    },
    "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 ザーンの作成」](#)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 ザーンのみに属することができます。

NSX-T では、異なるトランスポート ザーンに属する仮想マシン同士の接続は許可されません。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、NSX-T ホストスイッチ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 がその VLAN アップリンク用に使用します。NSX Edge が VLAN トランスポート ザーンに追加されると、VLAN ホストスイッチが NSX Edge にインストールされます。

ホストスイッチによって分散論理ルーター 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 ゾーンを作成する際には、そのトランスポート ゾ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、ホストスイッチの名前を指定する必要があります。ホストスイッチ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-mgr>>) にログインします。
- 2 [ファブリック (Fabric)] > [トランスポート ゾーン (Transport Zones)] の順に選択し、[追加 (Add)] をクリックします。
- 3 トランスポート ゾーン、ホストスイッチ名、トラフィック タイプ（オーバーレイまたは VLAN）を入力します。
次はその例です。

TRANSPORT ZONES			
<div> + ADD EDIT DELETE ACTIONS COLUMNS </div>			
<input type="checkbox"/> Transport Zone	ID	Traffic Type	Host Switch Name
<input type="checkbox"/> tz-overlay	efd7...a9ec	Overlay	overlay-hostswitch
<input type="checkbox"/> tz-vlan	9b66...b416	VLAN	vlan-uplink-hostswitch

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できるノードです。

KVM ホストの場合は、ホストスイッチ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できます。ESXi ホストの場合は、常に NSX Manager でホストスイッチが設定されます。

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 エージェントは新しい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 (Fabric)] > [ホスト (Hosts)] 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 (ホストスイッチ プロファイル) が設定されている必要があります。デフォルトのアップリンク プロファイルを使用することもできます。
- IP アドレス プールが設定されているか、ネットワーク展開で DHCP が使用可能である必要があります。
- ホスト上で 1 個以上の未使用の物理 NIC が使用可能である必要があります。

手順

- 1 ブラウザから NSX Manager (<https://<nsx-mgr>>) にログインします。
- 2 [ファブリック (Fabric)] > [ノード(Nodes)] > [トランスポート ノード (Transport Nodes)] を選択し、[追加 (Add)] をクリックします。
- 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 (オプション) 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。
- 6 (オプション) KVM ノードの場合は、ホストスイッチのタイプを選択します。

オプション	説明
標準	NSX Manager でホストスイッチが作成されます。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	ホストスイッチはすでに設定されています。

KVM 以外のノードの場合、ホストスイッチのタイプは常に [標準 (Standard)] になります。

- 7 標準ホストスイッチの場合は、次のホストスイッチ情報を入力または選択します。
 - ホストスイッチ名。そのノードが属するトランスポート ゾーンのホストスイッチ名と同じ名前である必要があります。
 - アップリンク プロファイル。
 - IP アドレスの割り当て。[DHCP を使用 (Use DHCP)]、[IP アドレス プールを使用 (Use IP Pool)]、または [固定 IP リストを使用 (Use Static IP List)] を選択できます。[固定 IP リストを使用 (Use Static IP List)] 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。

- 物理 NIC の情報。

重要: 物理 NIC がすでに使用されていないことを確認します（標準の仮想スイッチまたは vSphere 分散スイッチなどが使用していることがあります）。すでに使用されてる場合、トランスポート ノードは [部分的成功 (partial success)] 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。

Add Transport Node

Name: *

comp-02b

Node: *

comp-02b - 192.168.210.54

Transport Zones:

tz-overlay

Host Switch Type: *

☒ Standard
 ☐ Preconfigured

New Node Switch

Host Switch Name: *

overlay-hostswitch

Uplink Profile: *

uplinkProfile1

IP Assignment: *

Use IP Pool

IP Pool: *

ip-pool-1

OR Create and Use a new IP Pool

Physical NICs:

vmnic1

uplink-1

Save

Cancel

- 事前設定済みホストスイッチの場合は、次のホストスイッチ情報を入力します。

- ホストスイッチの外部 ID。そのノードが属するトランスポート ゾーンのホストスイッチ名と同じ ID である必要があります。
- VTEP 名。

ホストをトランスポート ノードとして追加した後、NSX Controller へのホスト接続が CLOSE_WAIT 状態から確立された状態に変わります。 `esxcli network ip connection list | grep 1234` コマンドでこれを確認できます。

```
# 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 [ESTABLISHED] -
```

トランスポート ノードは、GET `https://<nsx-mgr>/api/v1/transport-nodes/<node-id>` API 呼び出しで確認できます。

```
{
  "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
}
```

トランスポート ノードの作成に成功すると、[LCP 接続 (LCP Connectivity)] が [アップ (Up)] に変わります ([ファブリック (Fabric)] > [ノード (Nodes)] > [ホスト (Hosts)])。この変更を確認するには、ブラウザ画面を更新します。

次のステップ

NSX Edge トランスポート ノードを作成します。[\[NSX Edge トランスポート ノードの作成\]](#) を参照してください。

トランスポート ノードのステータスの確認

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 NSX-T ホストスイッチを配置します。

手順

- 1 **esxcli network ip interface list** コマンドを使用して、ESXi 上の NSX-T ホストスイッチを確認します。

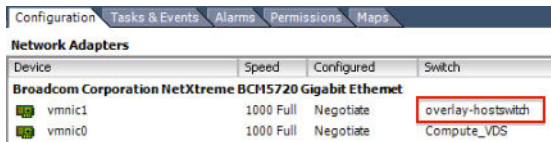
ESXi でのコマンドの出力には、Distributed Switch (VDS) の名前がついた vmk インターフェイス (vmk10 など) が含まれます。この Distributed Switch の名前は、トランスポート ゾーンとトランスポート ノードを設定する際に使用した名前です。

```
# 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 (Configuration)] > [ネットワーク アダプタ (Network Adapters)] を順に選択し、インストールされているホスト スイッチを確認できます。



Device	Speed	Configured	Switch
Broadcom Corporation NetXtreme BCM5720 Gigabit Ethernet			
vmnic1	1000 Full	Negotiate	overlay-hostswitch
vmnic0	1000 Full	Negotiate	Compute_VDS

NSX-T ホストスイッチを確認するための KVM のコマンドは、**ovs-vsctl show** です。KVM では、ホストスイッチの名前は **nsx-switch.0** と表示されます。トランスポート ノードの設定で使用し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"
```

- 2 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
...
```

3 API でステータスを確認します。

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"
}
```

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。ホストスイッチが含まれているノードは、トランスポート ノードとして機能できます。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。この手順は、NSX Edge をトランスポート ノードとして追加する方法を示しています。

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 (Fabric)] > [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（ホストスイッチ 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、ネットワーク展開で DHCP が使用可能である必要があります。
- 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 が使用可能である必要があります。

手順

- 1 ブラウザから NSX Manager (<https://<nsx-mgr>>) にログインします。
- 2 [ファブリック (Fabric)] > [ノード (Nodes)] > [トランスポート ノード (Transport Nodes)] を選択し、[追加 (Add)] をクリックします。
- 3 IP アドレス、ホストスイッチ名、アップリンク プロファイル、IP アドレス プール（または DHCP の選択）、および物理 NIC に関する情報を入力します。
 - NSX Edge トランスポート ノードの名前を入力します。
 - 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。
 - トランスポート ゾーンを選択します。通常、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。
 - ホストスイッチの名前を入力します。Edge スイッチ名（ホストスイッチ名と呼ばれることもあります）。Edge スイッチの名前は、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 プロファイルを選択します。
 - オーバーレイ ホストスイッチの IP アドレス プールを選択します。VLAN ホストスイッチの場合は、IP アドレス プールのフィールドを空のままにします。オーバーレイ ホストスイッチはアップリンク VLAN トラフィック専用であるため、オーバーレイ トンネル エンドポイントの IP アドレスは不要です。
 - 仮想 NIC およびアップリンクを選択します。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。

- アップリンクを選択します。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。

次はその例です。

Add Transport Node

Name: *

Node: *

Transport Zones:

overlay-hostswitch

Edge Switch Name: *

Uplink Profile: *

IP Pool:

Virtual NICs:

New Node Switch

Edge Switch Name: *

Uplink Profile: *

IP Pool:

Virtual NICs:

Add New Node Switch

Save
Cancel

4 [保存 (Save)] をクリックして終了します。

トランスポート ノードは、GET <https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>> API 呼び出しで確認できます。

```
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
  }
}

```

ステータス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。次はその例です。

```

{
  "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
  }
},
"pnict_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"
}

```

トランスポート ノードの作成に成功すると、[LCP 接続 (LCP Connectivity)] が [アップ (Up)] に変わります ([ファブリック (Fabric)] > [ノード (Nodes)] > [Edge])。この変更を確認するため、ブラウザ画面の再読み込みが必要となる場合があります。

次のステップ

NSX Edge ノードを 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、少なくとも 1 つの NSX Edge が常に使用可能になります。NAT で Tier-0 分散論理ルーターまたは Tier-1 分散論理ルーターを作成するには、それを NSX Edge クラスタと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分散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 (Fabric)] > [プロファイル (Profiles)] > [Edge クラスタ プロファイル (Edge Cluster Profiles)])。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 NSX Manager のユーザー インターフェイスで、[ファブリック (Fabric)] > [ノード (Nodes)] > [Edge クラスタ (Edge Clusters)] の順に選択します。
- 2 NSX Edge クラスタ名を入力します。
- 3 NSX Edge クラスタ プロファイルを選択します。
- 4 [編集 (Edit)] をクリックし、[物理 (Physical)] または [仮想 (Virtual)] を選択します。
「物理」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」は、仮想マシン/アプリケーションとして配置されている NSX Edge を意味します。
- 5 [使用可能 (Available)] 列から NSX Edge を選択し、右矢印をクリックして [選択済み (Selected)] 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-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行う方法について説明します。

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 のユーザー インターフェイスで、[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 API で **DELETE /api/v1/transport-node/<node-id>** を使用して、ホスト 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 NSX-T ホストスイッチが削除されます。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 のインストールおよびアップグレード サービス (install-upgrade) を有効にして開始します。

```
nsx-manager-1> set service install-upgrade enable
nsx-manager-1> start service install-upgrade
```

- 4 管理プレーンからホストを登録解除して、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 10 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 のユーザー インターフェイスで、[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。ホストの状態が良好な場合は、[ファブリック (Fabric)] > [ノード (Nodes)] > [ホスト (Hosts)] > [削除 (Delete)] を使用する際に、[NSX コンポーネントのアンインストール (Uninstall NSX Components)] オプションを必ず選択してください。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。

- **DELETE /api/v1/fabric/nodes/<node-id>** API を使用します。

- CLI を使用します。

- 1 管理用のサムプリントを取得します。

```
manager> get certificate api thumbprint
```

- 2 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 <MANAGER-USERNAME>
password <MANAGER-PASSWORD> thumbprint <MANAGER-THUMBPRINT>
```

- 3 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
```

- 4 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
```

- 5 ホスト上の仮想マシンをパワーオフします。
- 6 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-aggsservice -n nsx-da -n nsx-esx-datapath -n nsx-exporter -n nsx-host -n nsx-lldp -n nsx-mpa -n nsx-netcpa -n nsx-python-protobuf -n nsx-sfhc -n nsx-support-bundle-client -n nsxa -n nsxcli
```

RHEL では、`sudo yum remove <package-name>` コマンドを使用します。Ubuntu では、`apt-get remove <package-name>` コマンドを使用します。

いずれの場合も、ワイルドカードを使用して、NSX-T モジュールを選択します。

また、次のモジュールも削除します。

- Ubuntu の場合：tcpdump-ovs、nicira-ovs-hypervisor-node、python-openvswitch、openvswitch-*、libgoogle-glog0、libjson-spirit
- RHEL の場合：tcpdump-ovs、openvswitch、kmod-openvswitch、glog、json_spirit

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。