



NX-SDK

この章は、次の項で構成されています。

- [NX-SDK について（1 ページ）](#)
- [オンボックス（ローカル）アプリケーションについて（3 ページ）](#)
- [デフォルト Docker イメージ（3 ページ）](#)
- [NX-SDK に関する注意事項と制限事項（3 ページ）](#)
- [オフボックス（リモート）アプリケーション（4 ページ）](#)

NX-SDK について

Cisco NX-OS SDK（NX-SDK）は、自動化およびカスタム アプリケーションの作成（カスタムの生成など）のためのインフラストラクチャへのアクセスを合理化する C++ 抽象化およびプラグイン ライブラリ レイヤです。

- CLI
- Syslog
- イベント マネージャとエラー マネージャ
- アプリケーション間通信
- ハイ アベイラビリティ（HA）
- ルート マネージャ

NX-SDK を使用したアプリケーション開発には、C++、Python、または Go（一部の例外あり）を使用できます。

要件

NX-SDK では次の要件があります。

- Docker

- Linux 環境（Ubuntu 14.04 または Centos 6.7 のいずれか）。提供されている NX-SDK Docker コンテナを使用することをお勧めします。詳細については、「[Cisco DevNet NX-SDK](#)」を参照してください。

ローカル（オン スイッチ）およびリモート（オフ スイッチ）アプリケーションのサポート

NX-SDK を使用して開発されたアプリケーションは、NX-SDK が提供する Docker コンテナのスイッチから作成または開発されます。アプリケーションの作成後、アプリケーションを展開できる場所を柔軟に選択できます。

- ローカル（オンボックス）アプリケーションはスイッチ上で実行されます。詳細については、[オンボックス（ローカル）アプリケーションについて（3 ページ）](#)を参照してください。
- リモート（オフボックス）アプリケーションは、スイッチをオフにして実行されます。このオプションは NX-SDK 2.0 以降でサポートされており、アプリケーションをスイッチ以外の場所で実行するようにデプロイできます。詳細については、[リモートアプリケーションについて（5 ページ）](#)を参照してください。

関連情報

Cisco NX-SDK の詳細については、次にアクセスしてください。

- [Cisco DevNet NX-SDK](#)。versions.md リンク（[Cisco DevNet NX-SDK バージョン](#)）をクリックして、サポートされている各リリースの機能と詳細に関する情報を取得します。
- [NX-SDK Readme](#)

必要に応じて、シスコは NX-SDK の情報を github に追加します。

Go バインディングに関する考慮事項

Go バインディングは、NX-SDK のリリースと、アプリがローカルで実行されているかリモートで実行されているかに応じて、さまざまなレベルでサポートされます。

- NX-SDK リモート アプリケーションのすべてのバージョンの Go バインディングは、EFT 前の品質です。
- ローカル NX-SDK 2.0 アプリケーションの Go バインディングは、EFT 前です。
- ローカル NX-SDK 1.7.5 以前のアプリケーションの Go バインディングがサポートされていません。

詳細については、「[NX-SDK アプリケーションの GO バインディング](#)」を参照してください。

オンボックス（ローカル）アプリケーションについて

オンボックス（ローカル）アプリケーションでは、NX-SDK をインストールし、選択したサポート対象言語でアプリケーションをビルドし、スイッチにインストールできる .rpm ファイルとしてアプリをパッケージ化し、スイッチにアプリケーションをインストールして実行します。。.rpm ファイルは手動で生成することも、自動生成することもできます。

アプリケーション開発は、NX-SDK が提供するコンテナで行われます。ローカルアプリケーションには、リモートアプリケーションとは異なるコンテナとツールを使用します。詳細については、[デフォルト Docker イメージ（3 ページ）](#) を参照してください。

ローカルアプリケーションのビルド、インストール、および実行については、[Cisco DevNet NX-SDK](#) を参照してください。

デフォルト Docker イメージ

NX-SDK には、ローカルまたはリモートで使用するための次の Docker イメージとツールがデフォルトに含まれています。

使用方法	目次
スイッチの場合	Cisco ENXOS SDK クロスコンパイル用の Wind River Linux（WRL） ツールチェーン 多言語バインディングツールキット NX-SDK 1.75 以降、Go コンパイラ
オフスイッチ（リモート）	事前にビルドされた libnxsdk.so を備えた NX-SDK 多言語バインディングツールキット Go コンパイラ RapidJSON リモート API サポートのための gRPC

詳細については、<https://github.com/CiscoDevNet/NX-SDK#readme>https://github.com/CiscoDevNet/NX-SDK/tree/master/readmes/nxsdk_docker_images.md を参照してください。

NX-SDK に関する注意事項と制限事項

NX-SDK には、アプリケーションをローカル（オンボックス）またはリモート（オフボックス）で実行するための使用上の注意事項と制限事項があります。

注意事項と制限事項については、[Cisco DevNet NX-SDK](#) の「役立つメモ」を参照してください。

- リモートアプリケーションの場合、SDK サーバーに接続するポートは 50002 ～ 50100 です。これらのポートが開いていて、他のサービスによって使用されていないことを確認してください。ポート 50051 は、内部アプリケーションによる使用がブロックされており、リモートアプリケーションでは使用できません。
- NX-SDK を使用したカスタム アプリケーションの開発には、次の制限が適用されます。
 - GNU C/C++ ツールチェーンと標準ライブラリを備えた現在の Linux 配布を使用してアプリケーションを開発する必要があります。
 - 提供されている Docker イメージでアプリケーションを開発することをお勧めします。Docker イメージと関連ツールを実行するには、最低 8 MB の空きメモリと 64 ビット ホストが必要です。「[デフォルト Docker イメージ（3 ページ）](#)」を参照してください。
- Cisco は、Bash でアプリケーションをビルドしてテストすることを推奨しています。それらを本番環境に展開する場合は、アプリケーションを RPM としてビルドし、NX-OS VSH（Vshell）で実行します。
- Cisco Nexus スイッチは、Bash および VSH で最大 32 のアプリケーションをサポートします。
- NX-SDK 2.0 の場合、Cisco Nexus スイッチで実行されているリモート NX-SDK アプリケーションおよび NX-SDK サーバーには 1 対 1 の制限があります。
- 最大 10 個のリモート NX-SDK アプリケーションをスイッチで同時に実行できます。
- NX-SDK サーバーは、CLI で構成されたアプリケーションからのリモート要求のみを受け入れ、ランダムなアプリケーションからは受け入れません。
- エラーが発生すると、SDK サーバーは例外をスローします。
- 一部の API は、リモートアプリケーションでは実行されません。詳細については、API ドキュメントを参照してください。
- リモートクライアントライブラリコードはオープンソースではないため、NX-SDK リモートアプリは NX-SDK リモート Docker コンテナで実行する必要があります。自分の OS でリモート NX-SDK を使用するには、使用している OS とコンパイラのバージョンを github でリクエストします。Cisco は、バージョンに基づいてリモート libnxsdk.so を生成できます。

オフボックス（リモート）アプリケーション

NX-SDK2.0 について

NX-SDK バージョン 2.0 は、開発者が必要な場所でアプリケーションを実行できる実行環境の柔軟性を可能にします。このバージョンの NX-SDK では、アプリケーションは引き続きコンテナ内のスイッチをオフにして開発されますが、たとえばクラウドなどで、アプリケーションをスイッチ上またはスイッチ外のいずれかで実行できます。

NX-SDK 2.0 には次のような利点があります。

- スイッチをお客様の環境に簡単に統合できます。
- スイッチがデータセンター、パブリッククラウド、プライベートクラウドでシームレスに動作できるようにする拡張性。
- スイッチ レベルのリソースでの変更がアプリケーションの変更または書き換えを必要としないように、スイッチ リソースから顧客アプリを切り離します。
- アプリケーションがリンクするための使いやすい API を備えた単一のライブラリ。これにより、スイッチの相互作用が簡素化され、アプリケーションをより簡単に記述およびデバッグできる高水準言語で記述できます。
- リモート サービスの実行は、オンボックス アプリケーションよりも安全です。

詳細については、https://github.com/CiscoDevNet/NX-SDK/blob/master/readmes/NXSDK_in_NXOS.md[NX-SDK アプリケーションのリモート展開（Deploying NX-SDK Applications Remotely）] を参照してください。

リモート アプリケーションについて

リモート アプリケーションは、Cisco Nexus スイッチではない別のスイッチに置くことができます。リモートまたはオフボックスのアプリケーションは、NX-SDK レイヤを介して呼び出し、スイッチと対話して情報の読み取り（取得）または情報の書き込み（設定）をします。

ローカルとリモートの両方の NX-SDK アプリケーションは同じ API を使用するため、NX-SDK アプリケーションをオンボックスまたはオフボックスで柔軟に展開できます。

リモートで実行するには、アプリケーションが特定の要件を満たしている必要があります。詳細については、https://github.com/CiscoDevNet/NX-SDK/blob/master/readmes/NXSDK_in_NXOS.md[NX-SDK アプリケーションのリモート展開（Deploying NX-SDK Applications Remotely）] を参照してください。

2.0 より前の NX-SDK アプリケーションの後方互換性

NX-SDK 2.0 には、NX-SDK v1.75 アプリケーションの開発方法に応じて、条件付きの後方互換性があります。

- 通常、NX-SDK は、NX-SDK 2.0 より前に作成したアプリのリモート実行をサポートしており、アプリを完全に書き直す必要はありません。代わりに、API 呼び出しを変更するために変更せずに、同じアプリを再利用できます。新しいNX-SDK 2.0モデルで古いアプリをサポートするには、API 呼び出しで IP およびポートのパラメータを提供する必要があります。これらのパラメータは NX-SDK 1.75 以前では使用できませんが、アプリが SDK サーバにエクスポートできる環境変数として IP アドレスとポート情報を追加できます。
- ただし、NX-SDK 2.0 より前のアプリの下位互換性がサポートされていない場合があります。古いアプリの一部の API は、リモートでの実行をサポートしていないか、実行できない可能性があります。この場合、API は例外をスローできます。元のアプリケーションに対する例外処理の完全性と堅牢性によっては、アプリケーションが予期しない動作をする可能性があります。最悪の場合、クラッシュする可能性があります。

詳細については、https://github.com/CiscoDevNet/NX-SDK/blob/master/readmes/NXSDK_in_NXOS.md[NX-SDK アプリケーションのリモート展開（Deploying NX-SDK Applications Remotely）] を参照してください。

NX-SDK セキュリティ

NX-OS 9.3（1）以降、NX-SDK 2.0 は次のセキュリティ機能をサポートしています。

- セッションセキュリティ。リモートアプリケーションは、トランスポート レイヤー サービス（TLS）を介してスイッチ上の NX SDK サーバに接続し、アプリケーションとスイッチの NX SDK サーバ間に暗号化されたセッションを提供できます。
- サーバ証明書のセキュリティ。Cisco NX-OS 9.3（1）を使用した新しいスイッチ展開の場合、NX-SDK サーバは 1 日限りの一時証明書を生成して、カスタム証明書をインストールするのに十分な時間を提供します。

たとえば、以前の NX-SDK バージョンから NX-SDK 2.0 にアップグレードする場合など、NX-SDK サーバにカスタム証明書がすでにインストールされている場合、既存の証明書はアップグレード後も保持され、使用されます。
- API 書き込み呼び出し制御。NX-SDK 2.0 では、セキュリティプロファイルが導入されています。これにより、アプリケーションが NX-SDK サーバをどの程度制御できるかを制御するための事前定義されたポリシーを選択できます。セキュリティプロファイルに関する詳細情報を入手するには、[NX SDK 2.0 のセキュリティ プロファイル（6 ページ）](#) を参照します。

NX SDK 2.0 のセキュリティ プロファイル

以前のリリースでは、SDK バージョン 1.75 の API は、イベントのデータの読み取りと取得のみが許可されていました。Cisco NX-OS リリース 9.3(1) 以降、NX-SDK 2.0 は書き込みコールを含むさまざまなタイプの操作をサポートします。

アプリがスイッチを読み書きする機能は、セキュリティプロファイルを介して制御できます。セキュリティプロファイルは、スイッチで実行されているアプリケーションのサービスに付加されるオプションのオブジェクトです。セキュリティプロファイルは、スイッチに書き込むアプリケーションの機能を制御し、スイッチ機能を変更、削除、または構成するアプリケーションの機能を制御します。デフォルトでは、アプリケーションの書き込みは許可されていないため、アプリケーションごとに、スイッチへの書き込みアクセスを有効にするセキュリティプロファイルを作成する必要があります。

Cisco の NX-SDK は、次のセキュリティプロファイルを提供します。

プロファイル	説明	値
拒否	CLI の追加を除き、API 呼び出しがスイッチに書き込まれないようにします。	これはデフォルトプロファイルです。

プロファイル	説明	値
スロットル	スイッチを変更する API を許可しますが、指定された数の呼び出しまでのみ許可します。このセキュリティプロファイルは、スロットリングを適用して API 呼び出しの数を制御します。 アプリケーションは制限まで書き込むことができますが、制限を超えると書き込みが停止し、応答でエラーメッセージが送信されます。	スロットルは 50 回の API 呼び出しであり、スロットルは 5 秒後にリセットされます。
許可	スイッチを変更する API は制限なしで許可されます	

NX-SDK のセキュリティプロファイルの詳細については、[\[NX-SDK アプリケーションのセキュリティプロファイル \(Security Profiles for NX-SDK Applications\)\]](#) を参照してください。

アプリケーションの構築、インストール、および実行の詳細については、[CiscoDevNet NX-SDK](#) にアクセスしてください。

翻訳について

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