コマンドライン インターフェイスの概要

- Application Policy Infrastructure Controller について，1 ページ
- 設定オプション，1 ページ
- 管理対象オブジェクトの理解，2 ページ
- ファイル システムの理解，2 ページ

Application Policy Infrastructure Controller について

このガイドでは、Application Policy Infrastructure Controller（APIC）のコマンドライン インターフェイス（CLI）の使用方法について説明します。CLI は、標準の Bash コマンド言語インタプリタ シェルおよび APIC 用の一連のカスタム コマンドから構成されています。

API のクラス、メソッド、タイプの詳細については、Web ベースのアプリケーションである『*Cisco APIC Management Information Model Reference*』を参照してください。Application Policy Infrastructure Controller の機能と運用については、提供されているホワイト ペーパーおよび『*Cisco Application Centric Infrastructure Fundamentals*』を参照してください。

設定オプション

Cisco Application Policy Infrastructure Controller（APIC）は、次の設定オプションを備えています。

- オブジェクト モデル CLI を使用した直接設定：CLI オブジェクトの拡張機能を Bash シェルに使用して、管理対象オブジェクト（MO）および管理情報ツリー（MIT）を直接操作できます。このマニュアルでは、オブジェクト モデル CLI を使用した直接設定について説明します。
- NX-OS スタイルの CLI：Cisco APIC リリース 1.2 以降では、構成には NX-OS スタイルの CLI コマンドを使用できます。

（注） このドキュメントでは、APIC NX-OS スタイル の CLI インターフェイスに関する情報を記載していません。詳細については、『*Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*』を参照してください。

- シェル スクリプト：Bash シェルを使用し、シェル スクリプトによって一部のタスクを自動化できます。Bash の詳細については、「GNU Bash シェルの理解」を参照してください。
- Python API：さらに広範な自動化が可能になります。Python API の詳細については、『*Cisco APIC Python SDK Reference*』 を参照してください。

（注） Cisco APIC リリース 1.0 から リリース 1.2 までは、オブジェクト モデル CLI が、SSH を使用して APIC にログインする時に表示されるデフォルトの CLI でした。Cisco APIC リリース 1.2 以降、デフォルトの CLI は、NX-OS スタイルの CLI です。オブジェクト モデル CLI は、最初の CLI プロンプトでの **Bash** コマンドの入力に使用できます。

管理対象オブジェクトの理解

APIC システムの設定とステータスは、管理対象オブジェクト（MO）の集合としてモデル化されます。MO は、一連の設定とプロパティを含む物理または論理エンティティの抽象的表現です。たとえば、サーバ、シャーシ、I/O カード、プロセッサなどは、MO として表現される物理エンティティです。また、リソース プール、ユーザ ロール、サービス プロファイル、ポリシーなどは、MO として表現される論理エンティティです。

実行時、すべての MO は管理情報ツリーと呼ばれるツリー構造に編成されます。これによって、システム内のすべての MO に構造化された一貫性のある方法でアクセスできるようになります。

ファイル システムの理解

管理情報ツリー（MIT）を構成する MO は階層的に編成されていて、APIC の管理に使用できます。各 MO は Linux ディレクトリとしてモデル化されます。このディレクトリには、すべての子 MO がサブディレクトリとして含まれ、すべてのプロパティを含む mo ファイルも格納されます。

以下にファイル システムの例を示します。local-users ディレクトリには 3 人のユーザ（admin、john、閲覧者）用のサブディレクトリがあります。

```
admin@apic1:local-users> pwd
/home/admin/aci/admin/aaa/security-management/local-users
admin@apic1:local-users> ls -al
total 3
drw-rw---- 1 admin admin 512 Apr 10 16:58 .
drw-rw---- 1 root  root  512 Apr  8 07:06 ..
drw-rw---- 1 root  root  512 Apr  8 07:06 admin
drw-rw---- 1 admin admin 512 Jan 28 20:16 john
-r--r----- 1 admin admin 197 Apr 10 16:58 summary
```

ロールベース アクセス コントロール（RBAC）では、他のユーザを管理する権限をユーザに与えることができます。この例では、admin と閲覧ユーザはルート（root）によって所有され、john は admin によって所有されています。

（注） このディレクトリには mo ファイルがありません。これは、このディレクトリ レベルには設定可能なプロパティが存在しないことを示しています。

```
admin@apic1:local-users> cd admin
admin@apic1:admin> pwd
/home/admin/aci/admin/aaa/security-management/local-users/admin
admin@apic1:admin> ls -al
total 4
drw-rw---- 1 admin admin 512 Jul 22 14:29 .
drw-rw---- 1 admin admin 512 Jul 22 14:29 ..
-rw-rw---- 1 admin admin 485 Jul 22 14:29 mo
drw-rw---- 1 admin admin 512 Jul 22 14:29 operational
drw-rw---- 1 admin admin 512 Jul 22 14:29 security-domains
drw-rw---- 1 admin admin 512 Jul 22 14:29 ssh-keys
-r--r----- 1 admin admin 493 Jul 22 14:29 summary
drw-rw---- 1 admin admin 512 Jul 22 14:29 user-certificates
```

GNU Bash シェルの理解

Bash（Bourne Again SHell）は、さまざまなオペレーティング システムでサポートされている UNIX シェル（コマンドライン インタプリタ）です。Bash インターフェイスを使用して、直接 APIC を設定したり、タスクを自動化する Bash シェル スクリプトを作成することができます。Bash はさまざまなコマンド ライン機能とスクリプト機能を備えています。

構文

Bash は sh 互換のコマンド言語インタプリタであり、標準入力やファイルから読み込んだコマンドを実行します。また、Bash には Korn および C（ksh および csh）シェルの有用な機能が組み込まれています。Bash は、最終的には、IEEE POSIX Shell and Tools 仕様（IEEE Working Group 1003.2）を忠実に実装することを目指しています。

Bash は、次のようなさまざまな機能をサポートしています。

- コマンドラインの編集
- サイズに制限がないコマンド履歴
- ジョブ制御
- シェル機能およびエイリアス
- サイズに制限がないインデックス配列
- 2 ～ 64 までのあらゆる数値を底とする整数演算

Bash シェルの詳細については、http://www.gnu.org/software/bash/bash.html を参照してください。

Bash 拡張機能

APIC には 次のような Bash シェルの拡張機能が含まれています。

ネットワークの命名規則

通常、ネットワーク オペレーティング システムでは、インターフェイス、ネットワーク アドレス、その他の設定を区切る文字としてフォワードスラッシュ（/）が使用されています。しかし、Bash シェルでは、ファイル名でのフォワード スラッシュの使用は制限されています。Bash ではエスケープ文字が用意されていますが、APIC ファイル システムでは区切り文字としてコロン（:）を使用することで、ネットワークの命名を簡易化しています。以下に例を上げて、この区切り文字の使用方法を説明します。

インターフェイス名

APIC Bash 拡張機能では、コロン（:）を使用してインターフェイス名を区切ります。たとえば、インターフェイス Ethernet 1/46 は **eth1:46** と記述されます。

次の例は、あるノードのインターフェイスの出力を示しています。

```
admin@apic1:physical-interfaces> pwd
/aci/fabric/inventory/fabric-pod-1/fabric-node-17/interfaces/physical-interfaces
admin@apic1:physical-interfaces> ls
eth1:1   eth1:17  eth1:24  eth1:31  eth1:39  eth1:46  eth1:53  eth1:60
eth1:10  eth1:18  eth1:25  eth1:32  eth1:4   eth1:47  eth1:54  eth1:7
eth1:11  eth1:19  eth1:26  eth1:33  eth1:40  eth1:48  eth1:55  eth1:8
eth1:12  eth1:2   eth1:27  eth1:34  eth1:41  eth1:49  eth1:56  eth1:9
eth1:13  eth1:20  eth1:28  eth1:35  eth1:42  eth1:5   eth1:57  summary
eth1:14  eth1:21  eth1:29  eth1:36  eth1:43  eth1:50  eth1:58
eth1:15  eth1:22  eth1:3   eth1:37  eth1:44  eth1:51  eth1:59
eth1:16  eth1:23  eth1:30  eth1:38  eth1:45  eth1:52  eth1:6
admin@apic1:physical-interfaces>
```

ネットワーク アドレスの命名

APIC Bash 拡張機能では、コロン（:）を使用してネットワーク アドレスを区切ります。たとえば、ネットワーク 192.168.1.0 とサブネット 255.255.255.0 は次のように記述されます。

```
192.168.1.0:255.255.255.0
```

コマンドの補完

APIC は、標準の Linux コマンドや「コマンド リファレンス」に記載されている APIC 固有のコマンドに対して、タブ補完機能をサポートしています。コマンドまたはオプションの一部を入力して最後に Tab キーを押すと、コマンドの完全形、またはその後に続く可能性があるキーワードや引数の選択肢が表示されます。

たとえば、次のように、Tab キーを使用して利用可能なディレクトリを表示することができます。

```
admin@apic1:aci> cd tenants/ <Tab>
common/ infra/  mgmt/
```

コマンド履歴

APIC CLI は Bash シェルの履歴機能をサポートしています。コマンド履歴を表示する方法として、**history** コマンド以外にも、上矢印または下矢印キーを使用することができます。

履歴を閲覧して適切なコマンドを再び呼び出し、Enter キーを押すと、履歴内のコマンドを再入力できます。また、コマンドを再度呼び出した後、実行する前にコマンドを変更することもできます。

また、Ctrl-r キーを押して、必要なコマンドが表示されるまでその一部を入力することで、以前のコマンドを直接検索できます。

その他のコマンド履歴機能など、Bash シェルの詳細については、次の URL を参照してください。
http://www.gnu.org/software/bash/bash.html

コマンドのヘルプ

CLI には、次の 2 つの形式のコンテキスト ヘルプがあります。

- インライン ヘルプ：いつでも Esc キーを 2 回押して、その時点のコマンド構文で使用可能なオプションを表示できます。プロンプトに何も入力しないで Esc キーを 2 回押すと、その時点のコマンドモードで使用可能なコマンドがすべて一覧表示されます。コマンドの一部を入力して Esc キーを 2 回押すと、コマンド構文のその時点の位置で使用可能なキーワードと引数がすべて一覧表示されます。
- man ページ：コマンドプロンプトで、**man** と入力し、その後ろにコマンドを入力するか、/aci の下にある管理対象オブジェクト（MO）のパスを入力すると、UNIX 形式の man ページを表示することができます。man ページは、すべてのコマンドやスコープに対して利用できるわけではありません。

マウント ポイント

APIC CLI には 3 つのマウント ポイント（aci、mit、debug）があります。 以下のセクションで、マウント ポイントについて詳しく説明します。

APIC にログインすると、マウント ポイント aci、debug、mit がデフォルト ディレクトリとして表示されます。

```
admin@apic1:~> ls
aci  debug  mit
```

（注） 各ファイル システムへのリンクは、各ユーザのホーム ディレクトリに置かれています。

以下のセクションで、マウント ポイントについて詳しく説明します。

aci マウント ポイント

aci ファイル システムは、MO とプロパティを対話型ユーザ セッション向けの簡易形式にまとめます。aci マウント ポイントは APIC のプライマリ CLI インターフェイスであり、大部分のユーザを対象としています。

mit マウント ポイント

上級ユーザは管理情報ツリー（MIT）ファイル システムを使用し、MIT 内の MO を直接表示して設定することができます。mit ファイル システムのディレクトリ構造は aci と同じですが、MO はネイティブ MIT オブジェクトとして表示されます。

たとえば、mit マウント ポイントでは、管理ユーザは次のように表示されます。

```
admin@apic1:user-admin> pwd
/mit/uni/userext/user-admin
admin@apic1:user-admin> ls -ltr
total 4
drw-rw---- 1 root  root  512 Jan 27 15:08 userdomain-all
drw-rw---- 1 root  root  512 Jan 27 15:08 userdata
-r--r----- 1 root  root  665 Jan 27 15:08 mo
drw-rw---- 1 admin admin 512 Jan 28 17:56 history
drw-rw---- 1 admin admin 512 Jan 28 17:56 faults
```

（注） mit マウント ポイントは、MO の設定を十分に理解している上級ユーザを対象としています。

debug マウント ポイント

debug マウント ポイントでは、複数の APIC デバイス、リーフ デバイス、スパイン デバイスにわたって設定を表示してデバッグできます。debug マウント ポイントは、上級ユーザによるトラブルシューティング用です。

ロールベース アクセス コントロール

ロールベース アクセス コントロール（RBAC）では、ユーザにロールを割り当てることで、デバイス操作のアクセスを制限できます。アクセスが必要なユーザだけにアクセスを許可するように、カスタマイズすることが可能です。

権限とセキュリティの適用

ロールベース アクセス コントロール（RBAC）を使用すると、一連の権限を備えたロールを作成してユーザに割り当て、ユーザの権限を制御することができます。RBAC では、直接的に権限を設定するのではなく、ロールを割り当てることでユーザに権限を適用できます。

APIC CLI 内では、管理対象オブジェクト（MO）など、管理情報ツリー（MIT）の特定部分を操作する権限をユーザに付与することができます。

次の例は、APIC CLI 内の RBAC 権限を表示する **ls** コマンドの使用方法を示しています。このコマンドの出力は、ファイル、UNIX のファイル読み取り/書き込み/実行権限、およびファイルが最後に変更された日時を表示します。

```
admin@apic1:user-admin> ls -al
total 4
drw-rw---- 1 admin admin 512 Jul 22 14:25 .
drw-rw---- 1 admin admin 512 Jul 22 14:25 ..
-rw-rw---- 1 admin admin 421 Jul 22 14:25 mo
-r--r----- 1 admin admin 608 Jul 22 14:25 summary
drw-rw---- 1 admin admin 512 Jul 22 14:25 userdata
drw-rw---- 1 admin admin 512 Jul 22 14:25 userdomain-all
```

ユーザ管理

デフォルトでは、各ユーザにホームディレクトリ（/home/<username>）が与えられます。このディレクトリで、ユーザはサブディレクトリとファイルを作成することができます。/home/<username> 内で作成されたファイルはデフォルトの umask パーミッションを継承し、本人と管理者（admin）のみがアクセスできます。

ユーザは初めてログインしたときに、ファイルを保存する *userid* ディレクトリ（/home/jsmith など）を作成することをお勧めします。作成すると、APIC は、その */userid* ディレクトリをユーザのホーム ディレクトリとして扱うようになります。