Embedded Event Manager の設定

この章では、Embedded Event Manager（EEM）を設定して Cisco NX-OS デバイス上のクリティカル イベントを検出し、対処する方法について説明します。

EEM について

EEM はデバイス上で発生するイベントをモニタし、設定に基づいて各イベントの回復またはトラブルシューティングのためのアクションを実行します。

EEM は次の 3 種類の主要コンポーネントからなります。

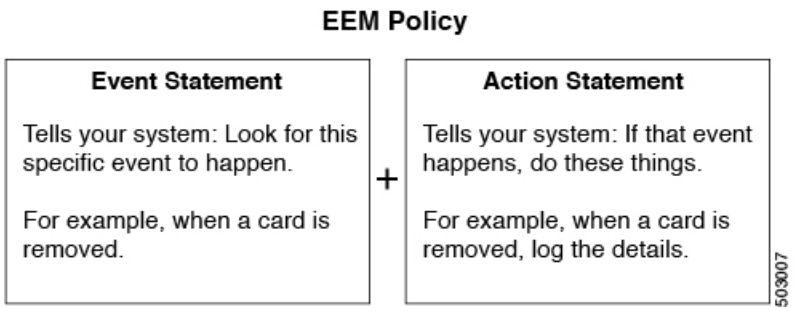
- イベント文：別の Cisco NX-OS コンポーネントからモニタし、アクション、回避策、または通知が必要になる可能性のあるイベント。
- アクション文：CLI コマンドの実行、Smart Call Home 機能を使用した電子メールの送信、インターフェイスの無効化など、イベントから回復するために EEM が実行できるアクション。
- ポリシー：イベントのトラブルシューティングまたはイベントからの回復を目的とした 1 つまたは複数のアクションとペアになったイベント。

ポリシー

EEM ポリシーは、イベント文および 1 つまたは複数のアクション文からなります。イベント文では、探すイベントとともに、イベントのフィルタリング特性を定義します。アクション文では、イベントの発生時に EEM が実行するアクションを定義します。

この図は、EEM ポリシーの基本的な 2 種類の文を示します。

図 *1: EEM* ポリシー文



コマンドライン インターフェイス（CLI）または VSH スクリプトを使用して EEM ポリシーを設定できます。

EEM からデバイス全体のポリシー管理ビューが得られます。スーパーバイザ上で EEM ポリシーを設定すると、EEM がイベント タイプに基づいて、正しいモジュールにポリシーをプッシュします。EEM はモジュール上でローカルに、またはスーパーバイザ上で（デフォルトのオプション）、発生したイベントに対応するアクションを実行します。

EEM はスーパーバイザ上でイベント ログを維持します。

Cisco NX-OS には、設定済みのさまざまなシステム ポリシーがあります。これらのシステム ポリシーでは、デバイスに関連する多数の一般的なイベントおよびアクションが定義されています。システム ポリシー名は、2 個の下線記号（__）から始まります。

使用するネットワークに合わせてユーザ ポリシーを作成できます。ユーザ ポリシーを作成すると、そのポリシーと同じイベントに関連するシステム ポリシー アクションが EEM によって発生したあと、ユーザ ポリシーで指定したアクションが行われます。

一部のシステム ポリシーは上書きすることもできます。設定した上書き変更がシステム ポリシーの代わりになります。イベントまたはアクションの上書きが可能です。

設定済みのシステム ポリシーを表示して、上書き可能なポリシーを判断するには、**show event manager system-policy** コマンドを使用します。

（注） **show running-config eem** コマンドを使用して、各ポリシーのコンフィギュレーションを確認してください。イベント文が指定されていて、アクション文が指定されていない上書きポリシーを設定した場合、アクションは開始されません。また、障害も通知されません。

（注） 上書きポリシーには、必ずイベント文を指定します。上書きポリシーにイベント文が含まれていないと、システム ポリシーで可能性のあるイベントがすべて上書きされます。

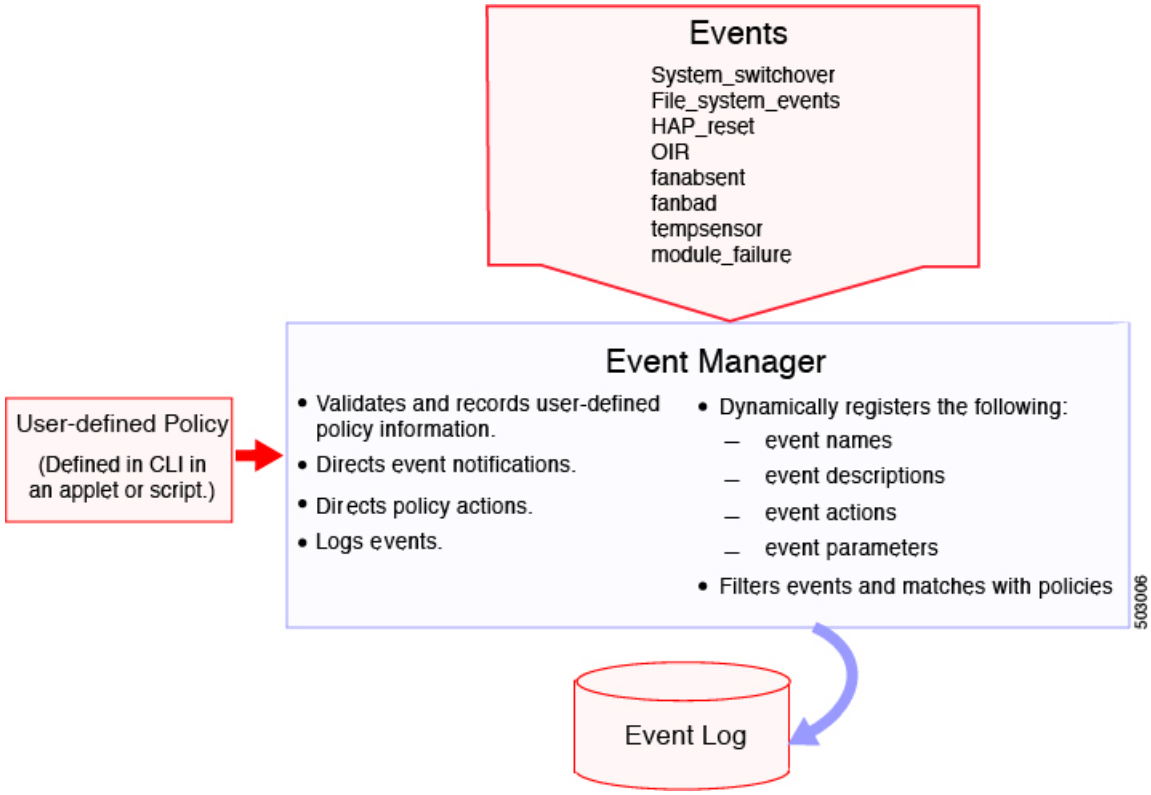
イベント文

イベントは、回避、通知など、何らかのアクションが必要なデバイス アクティビティです。これらのイベントは通常、インターフェイスやファンの誤動作といったデバイスの障害に関連します。

EEM ではイベント フィルタを定義して、クリティカル イベントまたは指定された時間内で繰り返し発生したイベントだけが関連付けられたアクションのトリガーになるようにします。

この図は、EEM によって処理されたイベントを示します。

図 *2 : EEM* の概要



イベント文では、ポリシー実行のトリガーになるイベントを指定します。複数イベント トリガーを設定できます。

EEM はイベント文に基づいてポリシーをスケジューリングし、実行します。EEM はイベントおよびアクション コマンドを検証し、定義に従ってコマンドを実行します。

（注） 発生したイベントでデフォルトのアクションを処理できるようにする場合は、event-default アクション文を許可して EEM ポリシーを設定する必要があります。

アクション文

アクション文では、ポリシーによって実行されるアクションを記述します。各ポリシーに複数のアクション文を設定できます。ポリシーにアクションを関連付けなかった場合、EEM はイベント観察を続けますが、アクションは実行されません。

EEM がアクション文でサポートするアクションは、次のとおりです。

- CLI コマンドの実行。
- カウンタのアップデート。
- 例外の記録。
- モジュールの強制的シャットダウン
- デバイスをリロードします。
- 電力のバジェット超過による特定モジュールのシャットダウン。
- Syslog メッセージの生成。
- Call Home イベントの生成。
- SNMP 通知の生成。
- システム ポリシー用デフォルト アクションの使用。

（注） EEM は、合計 1024 文字までの、完全なアクション CLI リストのみを処理できます。さらにアクションが必要な場合は、同じトリガーを持つ新しい冗長アプレットとして定義する必要があります。

（注） 発生したイベントでデフォルトのアクションを処理できるようにする場合は、デフォルトのアクションを許可する EEM ポリシーを設定する必要があります。たとえば、match 文で CLI コマンドを照合する場合、EEM ポリシーに event-default アクション文を追加する必要があります。この文がないと、EEM では CLI コマンドを実行できません。

（注） ユーザ ポリシーまたは上書きポリシーの中に、相互に否定したり、関連付けられたシステム ポリシーに悪影響を与えたりするようなアクション文がないかどうかを確認してください。

VSH スクリプト ポリシー

テキスト エディタを使用し、VSH スクリプトでポリシーを作成することもできます。このようなポリシーにも、他のポリシーと同様、イベント文およびアクション文（複数可）を使用します。また、これらのポリシーでシステム ポリシーを補うことも上書きすることもできます。VSH スクリプト ポリシーの作成後、そのポリシーをデバイスにコピーしてアクティブにします。

環境変数

すべてのポリシーに使用できる、EEM の環境変数を定義できます。環境変数は、複数のポリシーで使用できる共通の値を設定する場合に便利です。たとえば、外部電子メール サーバの IP アドレスに対応する環境変数を作成できます。

パラメータ置換フォーマットを使用することによって、アクション文で環境変数を使用できます。

この例では、「EEM action」というリセット理由を指定し、モジュール 1 を強制的にシャットダウンするアクション文の例を示します。

```
switch (config-eem-policy)# action 1.0 forceshut module 1 reset-reason "EEM action."
```

シャットダウンの理由に default-reason という環境変数を定義すると、次の例のように、リセット理由を環境変数に置き換えることができます。

```
switch (config-eem-policy)# action 1.0 foreshut module 1 reset-reason $default-reason
```

この環境変数は、任意のポリシーで再利用できます。

EEM イベント相関

イベントの組み合わせに基づいて EEM ポリシーをトリガーできます。まず、**tag** キーワードを使用して EEM ポリシーに複数のイベントを作成し区別します。次に、一連のブール演算子（**AND**、**OR**、**ANDNOT**）を使用して、回数および時間をもとに、カスタム処理をトリガーするこれらのイベントの組み合わせを定義できます。

高可用性

Cisco NX-OS は、EEM のステートレス リスタートをサポートします。リブートまたはスーパーバイザ スイッチオーバーの後、Cisco NX-OS は実行コンフィギュレーションを適用します。

仮想化のサポート

アクションまたはイベントがすべて表示されるわけではありません。ポリシーを設定するには、network-admin の権限が必要です。

EEM の前提条件

EEM の前提条件は、次のとおりです。

- EEM を設定するには、network-admin のユーザ権限が必要です。

EEM の注意事項と制約事項

EEM 設定時の注意事項と制約事項は次のとおりです。

- 設定可能な EEM ポリシーの最大数は 500 です。
- ユーザポリシーまたは上書きポリシー内のアクション文が、相互に否定したり、関連付けられたシステムポリシーに悪影響を与えたりするようなことがないようにする必要があります。
- 発生したイベントでデフォルトのアクションを処理できるようにするには、デフォルトのアクションを許可する EEM ポリシーを設定する必要があります。たとえば、match 文で CLI コマンドを照合する場合、EEM ポリシーに event-default アクション文を追加する必要があります。この文がないと、EEM では CLI コマンドを実行できません。
- 同じクライアントからは 10 個のトリガーのみ（たとえば、vshd は「イベント cli」のクライアント、snmp は「イベント snmp」のクライアントなど）が 1 秒以内に発行されます。
- オプション **collect** のアクションは、常に event applet action 文の最初のアクションである必要があります。
- イベント ログの自動収集とバックアップには、次の注意事項があります。
  - デフォルトでは、スイッチのログ収集を有効にすると、サイズ、規模、コンポーネントのアクティビティに応じて、15 分から数時間のイベントログが利用できるようになります。
  - 長期間にわたる関連ログを収集できるようにするには、必要な特定のサービス/機能に対してのみイベントログの保持を有効にします。「単一サービスの拡張ログ ファイル保持の有効化」を参照してください。内部イベントログをエクスポートすることもできます。「外部ログ ファイル ストレージ」を参照してください。
  - トラブルシューティングを行うときは、内部イベントログのスナップショットを手動によりリアルタイムで収集することをお勧めします。「最近のログ ファイルのローカル コピーの生成」を参照してください。

- **show tech** コマンドを収集するように EEM ポリシーアクションを設定する場合は、同じアクションが再度呼び出される前に、**show tech** コマンドが完了するのに十分な時間を割り当ててください。
- オーバーライド ポリシーについては、次の点に注意してください。
  - イベント文が指定されていても、アクション文が指定されていない上書きポリシーを設定した場合、アクションは開始されません。また、障害も通知されません。
  - 上書きポリシーにイベント文が含まれていないと、システム ポリシーで可能性のあるイベントがすべて上書きされます。
- 正規コマンド式には、次のルールが適用されます。
  - すべての正規表現は、Portable Operating System Interface for uniX（POSIX）拡張標準に準拠している必要があります。
  - すべてのキーワードを展開する必要があります。
  - 引数の置換には * 記号のみを使用できます。
- EEM イベント相関については、次の点に注意してください。
  - EEM イベント相関はスーパーバイザ モジュールだけでサポートされます。
  - EEMイベント相関は、単一ポリシー内の別のモジュール間ではサポートされません。
  - EEM イベント相関は 1 つのポリシーに最大 4 つのイベント文をサポートします。イベントタイプは同じでも別でもかまいませんが、サポートされるイベントタイプは、cli、カウンタ、モジュール、モジュール障害、oir、snmp、syslog だけです。
  - EEM イベント相関はシステムのデフォルト ポリシーを上書きしません。
- 複数のイベント文が EEM ポリシーに存在する場合は、各イベント文に **tag** キーワードと一意な tag 引数が必要です。
- デフォルトアクション実行は、タグ付きのイベントで設定されているポリシーではサポートされません。
- Python から EEM を呼び出すことができます。Python の詳細については、『Cisco Nexus 9000 シリーズ NX-OS プログラマビリティ ガイド』を参照してください。
- Cisco NX-OS リリース 10.3(1)F 以降、デフォルトの自動収集はシステム スイッチオーバーでサポートされていません。システムの切り替え時に、新しいアクティブ スーパーバイザで bloggerd auto-collect コマンドを再実行して、それぞれのコンポーネントの自動収集を有効にしてください。
- Cisco NX-OS リリース 10.3(3)F 以降、デフォルトのブロガー自動収集は、adjmgr、cts、l2fm、および vmtracker でサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、デフォルトのブロガー自動収集は、ipqosmgr、aclqos、cfs、ethport、feature-mgr、icam、interface manager、lacp、m2rib、mfdm、nbm、ngoam、

nve、port-channel、qos、sla_responder、sla_sender、sla_twamp、smm、spm、sysmgr、および vpc でサポートされます。

- 構成可能な最小消去時間が 0 時間から 48 時間に増加しました。
- 14 日を超えたファイルは、予約済みのブートフラッシュ領域（最大 5%）に関係なく、自動的に消去されます。
- 自動収集用に予約されたスペースが最大まで使用されている場合、ファイルの消去または手動によるファイルの削除によって予約済みスペースにまた空きができるまで、新しい自動収集は拒否されます。

EEM のデフォルト設定

この表では、EEM のデフォルト設定を一覧にしています。

パラメータ	デフォルト
システム ポリシー	アクティブ

EEM の設定

システム ポリシーに基づいて実行されるアクションを含むポリシーを作成できます。システム ポリシーに関する情報を表示するには、**show event manager system-policy** コマンドを使用します。

環境変数の定義

EEM ポリシーでパラメータとして機能する変数を定義できます。

手順の概要

1. **configure terminal**
2. **event manager environment** *variable-name variable-value*
3. （任意） **show event manager environment** {*variable-name* | **all**}
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

	コマンドまたはアクション	目的
ステップ 2	**event manager environment** *variable-name variable-value* 例： `switch(config)# event manager environment emailto "admin@anyplace.com"`	EEM 用の環境変数を作成します。*variable-name* は大文字と小文字を区別し、最大 29 文字の英数字を使用できます。*variable-value* には最大 39 文字の英数字を引用符で囲んで使用できます。
ステップ 3	（任意） **show event manager environment** {*variable-name* | **all**} 例： `switch(config)# show event manager environment all`	設定した環境変数に関する情報を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

CLI によるユーザ ポリシーの定義

CLI を使用して、デバイスにユーザ ポリシーを定義できます。

手順の概要

1. **configure terminal**
2. **event manager applet** *applet-name*
3. （任意） **description** *policy-description*
4. **event** *event-statement*
5. （任意） **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*}
6. **action** *number*[*.number2*] *action-statement*
7. （任意） **show event manager policy-state** *name* [**module** *module-id*]
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**event manager applet** *applet-name* 例：	EEM にアプレットを登録し、アプレット コンフィギュレーション モードを開始します。*applet-name*

	コマンドまたはアクション	目的
	`switch(config)# event manager applet monitorShutdown` `switch(config-applet)#`	は大文字と小文字を区別し、最大 29 文字の英数字を使用できます。
ステップ 3	（任意） **description** *policy-description* 例： `switch(config-applet)# description "Monitors interface shutdown."`	ポリシーの説明になるストリングを設定します。string には最大 80 文字の英数字を使用できます。ストリングは引用符で囲みます。
ステップ 4	**event** *event-statement* 例： `switch(config-applet)# event cli match "conf t ; interface * ; shutdown"`	ポリシーのイベント文を設定します。イベント文が複数ある場合、このステップを繰り返します。「イベント文の設定 （10 ページ）」を参照してください。
ステップ 5	（任意） **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*} 例： `switch(config-applet)# tag one or two happens 1 in 10000`	ポリシー内の複数のイベントを相互に関連付けます。 *occurs* 引数の範囲は 1 ～ 4294967295 です。*seconds* 引数の範囲は 0 ～ 4294967295 秒です。
ステップ 6	**action** *number*[.*number2*] *action-statement* 例： `switch(config-applet)# action 1.0 cli show interface e 3/1`	ポリシーのアクション文を設定します。アクション文が複数ある場合、このステップを繰り返します。「アクション文の設定 （16 ページ）」を参照してください。
ステップ 7	（任意） **show event manager policy-state** *name* [**module** *module-id*] 例： `switch(config-applet)# show event manager policy-state monitorShutdown`	設定したポリシーの状態に関する情報を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

イベント文の設定

イベント文を設定するには、アプレット コンフィギュレーション モードで次のいずれかのコマンドを使用します。

コマンド	目的
event application [**tag** *tag*] **sub-system** *sub-system-id* **type** *event-type* **例:** `switch(config-applet)# event application sub-system 798 type 1`	イベントの指定がサブシステムIDおよびアプリケーションイベントタイプに一致する場合に、イベントを発生させます。 *sub-system-id* と *event-type* の範囲は 1 ～ 4294967295 です。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 （注） このコマンドを使用するには、まず **feature evmed** コマンドを有効にして一般的なイベントディテクタを有効にする必要があります。
event cli [**tag** *tag*] **match** *expression* [**count** *repeats* | **time** *seconds*] **例:** `switch(config-applet)# event cli match "conf t ; interface * ; shutdown"`	正規表現と一致するコマンドが入力された場合に、イベントを発生させます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 *repeats* の範囲は 1 ～ 65000 です。time の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。
event counter [**tag** *tag*] **name** *counter* **entry-val** *entry* **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} [**exit-val** *exit* **exit-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**}] **例:** `switch(config-applet)# event counter name mycounter entry-val 20 gt`	カウンタが、開始演算子に基づいて開始のしきい値を超えた場合にイベントを発生させます。イベントはただちにリセットされます。任意で、カウンタが終了のしきい値を超えたあとでリセットされるように、イベントを設定できます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 *counter* name は大文字と小文字を区別し、最大 28 の英数字を使用できます。*entry* および *exit* の値の範囲は 0 ～ 2147483647 です。
event fanabsent [**fan** *number*] **time** *seconds* **例:** `switch(config-applet)# event fanabsent time 300`	秒数で設定された時間を超えて、ファンがデバイスから取り外されている場合に、イベントを発生させます。*number* の範囲はモジュールに依存します。*seconds* の範囲は 10 ～ 64000 です。

コマンド	目的
event fanbad [**fan** *number*] **time** *seconds* **例:** `switch(config-applet)# event fanbad time 3000`	秒数で設定された時間を超えて、ファンが故障状態の場合に、イベントを発生させます。*number* の範囲はモジュールに依存します。*seconds* の範囲は 10 ～ 64000 です。
event fib {**adjacency extra** | **resource tcam usage** | **route** {**extra** | **inconsistent** | **missing**}} **例:** `switch(config-applet)# event fib adjacency extra`	次のいずれかに対するイベントを発生させます。 • **adjacency extra**：ユニキャスト FIB に追加のルートがある場合。 • **resource tcam usage**：TCAM 使用率がいずれかの方向で 5 の倍数になるごとに。 • **route** {**extra** | **inconsistent** | **missing**}：ユニキャスト FIB でルートが追加、変更、または削除される場合。
event gold module {*slot* | **all**} **test** *test-name* [**severity** {**major** | **minor** | **moderate**}] **testing-type** {**bootup** | **monitoring** | **ondemand** | **scheduled**} **consecutive-failure** *count* **例:** `switch(config-applet)# event gold module 2 test ASICRegisterCheck testing-type ondemand consecutive-failure 2`	名前で指定されたオンライン診断テストが、設定された回数だけ連続して、設定された重大度で失敗した場合に、イベントを発生させます。*slot* の範囲は 1 ～ 10 です。*test-name* は設定されたオンライン診断テストの名前です。*count* の範囲は 1 ～ 1000 です。
event interface [**tag** *tag*] {**name** *interface slot/port* **parameter**} **例:** `switch(config-applet)# event interface ethernet 2/2 parameter`	カウンタが指定のインターフェイスに対して超えた場合に、イベントを発生させます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 （注） このコマンドを使用するには、まず **feature evmed** コマンドを有効にして一般的なイベントディテクタを有効にする必要があります。
event memory {**critical** | **minor** | **severe**} **例:** `switch(config-applet)# event memory critical`	メモリのしきい値を超えた場合にイベントを発生させます。メモリのしきい値の設定 （21 ページ）も参照してください。

コマンド	目的
event module [**tag** *tag*] **status** {**online** | **offline** | **any**} **module** {**all** | *module-num*} **例:** `switch(config-applet)# event module status offline module all`	指定したモジュールが選択された状態になったときにイベントを発生させます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。
event module-failure [**tag** *tag*] **type** *failure-type* **module** {*slot* | **all**} **count** *repeats* [**time** *seconds*] **例:** `switch(config-applet)# event module-failure type lc-failed module 3 count 1`	モジュールが設定された障害タイプになった場合に、イベントを発生させます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 *repeats* 範囲は 0 ～ 4294967295 です。*seconds* の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。
event none **例:** `switch(config-applet)# event none`	手動で指定されたイベントがないポリシー イベントを実行します。 （注） このコマンドを使用するには、まず **feature evmed** コマンドを有効にして一般的なイベントディテクタを有効にする必要があります。
event oir [**tag** *tag*] {**fan** | **module** | **powersupply**} {**anyoir** | **insert** | **remove**} [*number*] **例:** `switch(config-applet)# event oir fan remove 4`	設定されたデバイス構成要素（ファン、モジュール、または電源モジュール）がデバイスに取り付けられた場合、またはデバイスから取り外された場合に、イベントを発生させます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 任意で、ファン、モジュール、または電源モジュールの具体的な番号を設定できます。*number* の範囲は次のとおりです。 • ファン番号：モジュール依存 • モジュール番号：デバイス依存 • 電源モジュール番号：範囲は 1 ～ 3

コマンド	目的
event policy-default count *repeats* [**time** *seconds*] **例:** `switch(config-applet)# event policy-default count 3`	システム ポリシーで設定されているイベントを使用します。このオプションは、ポリシーを上書きする場合に使用します。 *repeats* の範囲は 1 ～ 65000 です。*seconds* の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。
event poweroverbudget **例:** `switch(config-applet)# event poweroverbudget`	電力バジェットが設定された電源モジュールの容量を超えた場合に、イベントを発生させます。
event snmp [**tag** *tag*] **oid** *oid* **get-type** {**exact** | **next**} **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} **entry-val** *entry* [**exit-comb** {**and** | **or**}] **exit-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} **exit-val** *exit* **exit-time** *time* **polling-interval** *interval* **例:** `switch(config-applet)# event snmp oid 1.3.6.1.2.1.31.1.1.1.6 get-type next entry-op lt 300 entry-val 0 exit-op eq 400 exit-time 30 polling-interval 300`	SNMP OID が、開始演算子に基づいて開始のしきい値を超えた場合にイベントを発生させます。イベントはただちにリセットされます。または任意で、カウンタが終了のしきい値を超えたあとでリセットされるように、イベントを設定できます。OID はドット付き10 進表記です。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 *entry* および *exit* の値の範囲は 0 ～ 18446744073709551615 です。time の範囲は 0 ～ 2147483647 秒です。interval の範囲は 1 ～ 2147483647 秒です。
event storm-control **例:** `switch(config-applet)# event storm-control`	ポート上のトラフィックが設定されたストーム制御しきい値を超えた場合に、イベントを発生させます。
event syslog [**occurs** *count*] {**pattern** *string* | **period** *time* | **priority** *level* | **tag** *tag*} **例:** `switch(config-applet)# event syslog period 500`	指定した syslog のしきい値を超えた場合にイベントを発生させます。カウントの範囲は 1 ～ 65000 で、時間の範囲は 1 ～ 4294967295 です。プライオリティの範囲は 0 ～ 7 です。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。

コマンド	目的
event sysmgr memory [**module** *module-num*] **major** *major-percent* **minor** *minor-percent* **clear** *clear-percent* **例:** `switch(config-applet)# event sysmgr memory minor 80`	指定したシステム マネージャのメモリのしきい値を超えた場合にイベントを発生させます。パーセンテージの範囲は 1 ～ 99 です。
event sysmgr switchover count *count* **time** *interval* **例:** `switch(config-applet)# event sysmgr switchover count 10 time 1000`	指定した switchover count が、指定した time interval を超えた場合にイベントを発生させます。switchover count の範囲は 1 ～ 65000 です。time interval の範囲は 0 ～ 2147483647 です。
event temperature [**module** *slot*] [*sensor-number*] **threshold** {**any** | **major** | **minor**} **例:** `switch(config-applet)# event temperature module 2 threshold any`	温度センサーが設定されたしきい値を超えた場合に、イベントを発生させます。sensor の範囲は 1 ～ 18 です。

コマンド	目的
event timer {**absolute time** *time* **name** *name* | **countdown time** *time* **name** *name* | **cron cronentry** *string* | **tag** *tag* | **watchdog time** *time* **name** *name*} **例:** `switch(config-applet)# event timer absolute time 100 name abtimer`	指定した時間に到達した場合に、イベントを発生させます。時間の範囲は 1 ～ 4294967295 です。 • **absolute time**：指定された絶対時刻が発生した場合に、イベントを発生させます。 • **countdown time**：指定された時間がゼロにカウントダウンされたときに、イベントを発生させます。タイマーはリセットされません。 • **cron cronentry**：CRON 文字列の指定が現在時刻に一致する場合に、イベントを発生させます。 • **watchdog time**：指定された時間がゼロにカウントダウンされたときに、イベントを発生させます。タイマーは、初期値に自動的にリセットされ、カウント ダウンが続行されます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 （注） このコマンドを使用するには、まず **feature evmed** コマンドを有効にして一般的なイベントディテクタを有効にする必要があります。
event track [**tag** *tag*] *object-number* **state** {**any** | **down** | **up**} **例:** `switch(config-applet)# event track 1 state down`	トラッキング対象オブジェクトが設定された状態になった場合に、イベントを発生させます。 **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 指定できる *object-number* の範囲は 1 ～ 500 です。

アクション文の設定

アクション文を設定するには、EEM コンフィギュレーション モードで次のいずれかのコマンドを使用します。

コマンド	目的
action *number*[.*number2*] **cli** *command1* [*command2...*] [**local**] **例:** `switch(config-applet)# action 1.0 cli "show interface e 3/1"`	設定された CLI コマンドを実行します。任意で、イベントが発生したモジュール上でコマンドを実行できます。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action *number*[.*number2*] **counter name** *counter* **value** *val* **op** {**dec** | **inc** | **nop** | **set**} **例:** `switch(config-applet)# action 2.0 counter name mycounter value 20 op inc`	設定された値および操作でカウンタを変更します。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 counter name は大文字と小文字を区別し、最大 28 の英数字を使用できます。*val* には 0 ～ 2147483647 の整数または置換パラメータを指定できます。
action *number*[.*number2*] **event-default** **例:** `switch(config-applet)# action 1.0 event-default`	関連付けられたイベントのデフォルトアクションを実行します。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action *number*[.*number2*] **forceshut** [**module** *slot* | **xbar** *xbar-number*] **reset-reason** *seconds* **例:** `switch(config-applet)# action 1.0 forceshut module 2 reset-reason "flapping links"`	モジュール、クロスバー、またはシステム全体を強制的にシャットダウンします。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 リセット理由は、引用符で囲んだ最大 80 文字の英数字ストリングです。
action *number*[.*number2*] **overbudgetshut** [**module** *slot*[-*slot*]] **例:** `switch(config-applet)# action 1.0 overbudgetshut module 3-5`	電力バジェット超過の問題により、1 つまたは複数のモジュールまたはシステム全体を強制的にシャットダウンします。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。

コマンド	目的
action *number*[*.number2*] **policy-default** **例:** `switch(config-applet)# action 1.0 policy-default`	上書きしているポリシーのデフォルトアクションを実行します。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action *number*[*.number2*] **publish-event** **例:** `switch(config-applet)# action 1.0 publish-event`	アプリケーション固有のイベントの発行を強制します。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action *number*[*.number2*] **reload** [**module** *slot*[*-slot*]] **例:** `switch(config-applet)# action 1.0 reload module 3-5`	1 つまたは複数のモジュールまたはシステム全体を強制的にリロードします。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action *number*[*.number2*] **snmp-trap** {[**intdata1** *data* [**intdata2** *data*]] [**strdata** *string*]} **例:** `switch(config-applet)# action 1.0 snmp-trap strdata "temperature problem"`	設定されたデータを使用して SNMP トラップを送信します。*number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 *data* 引数には、最大 80 桁の任意の数を指定できます。*string* には最大 80 文字の英数字を使用できます。
action *number*[*.number2*] **syslog** [**priority** *prio-val*] **msg** *error-message* **例:** `switch(config-applet)# action 1.0 syslog priority notifications msg "cpu high"`	設定されたプライオリティで、カスタマイズした syslog メッセージを送信します。*number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 *error-message* には最大 80 文字の英数字を引用符で囲んで使用できます。

（注） 発生したイベントでデフォルトのアクションを処理できるようにする場合は、デフォルトのアクションを許可する EEM ポリシーを設定する必要があります。たとえば、match 文で CLI コマンドを照合する場合、EEM ポリシーに event-default アクション文を追加する必要があります。この文がないと、EEM では CLI コマンドを実行できません。**terminal event-manager bypass** manager bypass コマンドを使用して、CLI での すべての EEM ポリシーを、CLI コマンドの実行と一致させることができます。

VSH スクリプトによるポリシーの定義

VSH スクリプトを使用してポリシーを定義できます。

始める前に

管理者の権限でログインしていることを確認します。

スクリプト名がスクリプト ファイル名と同じ名前であることを確認します。

ステップ 1 テキスト エディタで、ポリシーを定義するコマンド リストを指定します。
ステップ 2 テキスト ファイルに名前をつけて保存します。
ステップ 3 次のシステム ディレクトリにファイルをコピーします。bootflash://eem/user_script_policies

VSH スクリプト ポリシーの登録およびアクティブ化

VSH スクリプトで定義したポリシーを登録してアクティブにできます。

手順の概要

1. **configure terminal**
2. **event manager policy** *policy-script*
3. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**event manager policy** *policy-script* 例： `switch(config)# event manager policy moduleScript`	EEM スクリプト ポリシーを登録してアクティブにします。*policy-script* は大文字と小文字を区別し、最大 29 の英数字を使用できます。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

ポリシーの上書き

システム ポリシーは上書き可能です。

手順の概要

1. **configure terminal**
2. （任意） **show event manager policy-state** *system-policy*
3. **event manager applet** *applet-name* **override** *system-policy*
4. （任意） **description** *policy-description*
5. **event** *event-statement*
6. **action** *number action-statement*
7. （任意） **show event manager policy-state** *name*
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	（任意） **show event manager policy-state** *system-policy*
	例：
	`switch(config-applet)# show event manager policy-state __ethpm_link_flap`
	`Policy __ethpm_link_flap`
	`Cfg count : 5`
	`Cfg time interval : 10.000000 (seconds)`
	`Hash default, Count 0`
		上書きするシステム ポリシーの情報をしきい値を含めて表示します。システム ポリシー名を突き止めるには、**show event manager system-policy** コマンドを使用します。システム ポリシーについては、Embedded Event Manager システム イベントおよび設定例 を参照してください。

ステップ 3	**event manager applet** *applet-name* **override** *system-policy*
	例：
	`switch(config)# event manager applet ethport override __ethpm_link_flap`
	`switch(config-applet)#`
		システム ポリシーを上書きし、アプレット コンフィギュレーション モードを開始します。*applet-name* は大文字と小文字を区別し、最大 29 文字の英数字を使用できます。*system-policy* は、システム ポリシーの 1 つにする必要があります。

ステップ 4	（任意） **description** *policy-description* 例： `description “Overrides link flap policy.”`	ポリシーの説明になるストリングを設定します。string には最大 80 文字の英数字を使用できます。ストリングは引用符で囲みます。
ステップ 5	必須: **event** *event-statement* 例： `switch(config-applet)# event policy-default count 2 time 1000`	ポリシーのイベント文を設定します。

	コマンドまたはアクション	目的
ステップ 6	必須: **action** *number action-statement* 例： `switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."`	ポリシーのアクション文を設定します。 アクション文が複数ある場合、このステップを繰り返します。
ステップ 7	（任意） **show event manager policy-state** *name* 例： `switch(config-applet)# show event manager policy-state ethport`	設定したポリシーに関する情報を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

メモリのしきい値の設定

イベントを発生させるメモリしきい値を設定し、オペレーティングシステムがメモリを割り当てられない場合にプロセスを終了させるかどうかを設定できます。

始める前に

管理者の権限でログインしていることを確認します。

手順の概要

1. **configure terminal**
2. **system memory-thresholds minor** *minor* **severe** *severe* **critical** *critical*
3. （任意） **system memory-thresholds threshold critical no-process-kill**
4. （任意） **show running-config | include "system memory"**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**system memory-thresholds minor** *minor* **severe** *severe* **critical** *critical* 例：	EEM メモリ イベントを生成するシステム メモリしきい値を設定します。デフォルト値は次のとおりです。

	コマンドまたはアクション	目的
	`switch(config)# system memory-thresholds minor 60 severe 70 critical 80`	• マイナー - 85 • 深刻 - 90 • 重大 - 95 これらのメモリのしきい値を超えた場合、システムは次の syslog を生成します。 • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR ALERT RECOVERED • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE ALERT RECOVERED • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL ALERT RECOVERED
ステップ 3	（任意） **system memory-thresholds threshold critical no-process-kill** 例： `switch(config)# system memory-thresholds threshold critical no-process-kill`	メモリを割り当てられない場合もプロセスを終了しないようにシステムを設定します。デフォルト値では、最もメモリを消費するプロセスから終了できます。
ステップ 4	（任意） **show running-config | include "system memory"** 例： `switch(config-applet)# show running-config | include “system memory”`	システム メモリ設定に関する情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

EEM パブリッシャとしての syslog の設定

スイッチからの syslog メッセージをモニタできます。

（注） syslog メッセージをモニタする検索文字列の最大数は 10 です。

始める前に

EEM は、Syslog による登録に使用可能である必要があります。

Syslog デーモンが設定され、実行される必要があります。

手順の概要

1. **configure terminal**
2. **event manager applet** *applet-name*
3. **event syslog** [**tag** *tag*] {**occurs** *number* | **period** *seconds* | **pattern** *msg-text* | **priority** *priority*}
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**event manager applet** *applet-name*
	例：
	`switch(config)# event manager applet abc`
	`switch(config-applet)#`
		EEM にアプレットを登録し、アプレット コンフィギュレーション モードを開始します。

ステップ 3	**event syslog** [**tag** *tag*] {**occurs** *number* | **period** *seconds* | **pattern** *msg-text* | **priority** *priority*} 例： `switch(config-applet)# event syslog occurs 10`	syslog メッセージを監視し、ポリシーの検索文字列に基づいてポリシーを呼び出します。 • **tag** *tag* キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 • **occurs** *number* のキーワードと引数のペアは、発生回数を指定します。指定できる範囲は 1 ～ 65000 です。 • **period** *seconds* のキーワードと引数のペアは、発生回数を指定します。値の範囲は 1 ～ 4294967295 です。

	コマンドまたはアクション	目的
		• **pattern** *msg-text* のキーワードと引数のペアは、マッチさせる正規表現を指定します。パターンには、文字テキスト、環境変数、またはこの 2 つの組み合わせを含めることができます。文字列に空白が含まれる場合は引用符で囲みます。 • **priority** *priority* のキーワードと引数のペアは、syslog メッセージのプライオリティを指定します。このキーワードを指定しないと、すべての Syslog メッセージのプライオリティ レベルが「情報レベル」に設定されます。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

EEM の設定確認

EEM 設定情報を表示するには、次のいずれかの作業を実行します。

コマンド	目的
show event manager environment [*variable-name* | **all**]	イベント マネージャの環境変数に関する情報を表示します。
show event manager event-types [*event* | **all** | **module** *slot*]	イベントマネージャのイベント タイプに関する情報を表示します。
show event manager history events [**detail**] [**maximum** *num-events*] [**severity** {**catastrophic** | **minor** | **moderate** | **severe**}]	すべてのポリシーについて、イベント履歴を表示します。
show event manager policy-state *policy-name*	しきい値を含め、ポリシーの状態に関する情報を表示します。
show event manager script system [*policy-name* | **all**]	スクリプト ポリシーに関する情報を表示します。
show event manager system-policy [**all**]	定義済みシステム ポリシーに関する情報を表示します。
show running-config eem	EEM の実行コンフィギュレーションに関する情報を表示します。

コマンド	目的
show startup-config eem	EEM のスタートアップ コンフィギュレーションに関する情報を表示します。

EEM の設定例

モジュール 3 の中断のないアップグレード エラーのしきい値だけを変更することによって、__lcm_module_failure システム ポリシーを上書きする方法の例を示します。この例では、syslog メッセージも送信されます。その他のすべての場合、システム ポリシー __lcm_module_failure の設定値が適用されます。

```
event manager applet example2 override __lcm_module_failure
event module-failure type hitless-upgrade-failure module 3 count 2
action 1 syslog priority errors msg module 3 “upgrade is not a hitless upgrade!”
action 2 policy-default
```

__ethpm_link_flap システム ポリシーを上書きし、インターフェイスをシャットダウンする方法の例を示します。

```
event manager applet ethport override __ethpm_link_flap
event policy-default count 2 time 1000
action 1 cli conf t
action 2 cli int et1/1
action 3 cli no shut
```

CLI コマンドの実行を許可し、ユーザがデバイスでコンフィギュレーション モードを開始すると SNMP 通知を送る EEM ポリシーを作成する例を示します。

```
event manager applet TEST
event cli match "conf t"
action 1.0 snmp-trap strdata "Configuration change"
action 2.0 event-default
```

（注） EEM ポリシーに **event-default** アクション文を追加する必要があります。この文がないと、EEM では CLI コマンドを実行できません。

次に、EEM ポリシーの複数イベントを関連付け、イベント トリガーの組み合わせに基づいてポリシーを実行する例を示します。この例では、EEM ポリシーは、指定された syslog パターンのいずれかが 120 秒以内に発生したときにトリガーされます。

```
event manager applet eem-correlate
event syslog tag one pattern "copy bootflash:.* running-config.*”
event syslog tag two pattern “copy run start”
event syslog tag three pattern “hello”
tag one or two or three happens 1 in 120
action 1.0 reload module 1
```

最大障害しきい値に達すると、AsicMemory、FpgaRegTest、および L2ACLRedirect システム ポリシーによってスイッチのリロードが強制されます。次に、これらのポリシーのいずれかのデフォルト アクションを上書きし、代わりに syslog を発行する例を示します。

```
event manager applet gold override __fpgareg
action 1 syslog priority emergencies msg FpgaRegTest_override
```

次に、デフォルト ポリシーを上書きし、デフォルト アクションを実行する例を示します。

```
event manager applet gold_fpga_ovrd override __fpgareg
  action 1 policy-default
  action 2 syslog priority emergencies msg FpgaRegTest_override
```

（注） その他の設定例については、「Embedded Event Manager システム イベントおよび設定例」を参照してください。

イベント ログの自動収集とバックアップ

自動的に収集されたイベント ログは、スイッチのメモリにローカルに保存されます。イベント ログ ファイル ストレージは、一定期間ファイルを保存する一時バッファです。時間が経過すると、バッファのロールオーバーによって次のファイルのためのスペースが確保されます。ロールオーバーでは、先入れ先出し方式が使用されます。

Cisco NX-OS リリース 9.3(3) 以降、EEM は以下の収集およびバックアップ方法を使用します。

- 拡張ログ ファイルの保持
- トリガーベースのイベント ログの自動収集

拡張ログ ファイルの保持

Cisco NX-OS リリース 9.3 (3) 以降、すべての Cisco Nexus プラットフォーム スイッチは、少なくとも 8 GB のシステムメモリを備え、イベント ロギング ファイルの拡張保持をサポートします。ログ ファイルをスイッチにローカルに保存するか、外部コンテナを介してリモートに保存すると、ロールオーバーによるイベント ログの損失を削減できます。

すべてのサービスの拡張ログ ファイル保持のイネーブル化

拡張ログ ファイル保持は、スイッチで実行されているすべてのサービスに対してデフォルトで有効になっています。スイッチでログ ファイル保持機能がイネーブルになっていない場合（**no bloggerd log-dump** が設定されている場合）、次の手順を使用してイネーブルにします。

手順の概要

1. **configure terminal**

2. **bloggerd log-dump all**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**bloggerd log-dump all**
	例：
	`switch(config)# bloggerd log-dump all`
	`switch(config)#`
		すべてのサービスのログ ファイル保持機能をイネーブルにします。

例

```
switch# configure terminal
switch(config)# bloggerd log-dump all
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)#
```

すべてのサービスの拡張ログ ファイル保持の無効化

拡張ログ ファイル保持は、スイッチ上のすべてのサービスに対してデフォルトで有効になっています。スイッチのログ ファイル保持機能がすべてのサービスに対して有効になっている場合は、次の手順を実行します。

手順の概要

1. **configure terminal**
2. **no bloggerd log-dump all**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**no bloggerd log-dump all**
	例：
	`switch(config)# no bloggerd log-dump all`
	`switch(config)#`
		スイッチ上のすべてのサービスのログ ファイル保持機能を無効にします。

例

```
switch# configure terminal
switch(config)# no bloggerd log-dump all
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)#
```

単一サービスの拡張ログファイル保持の有効化

拡張ログファイル保持は、スイッチで実行されているすべてのサービスに対してデフォルトで有効になっています。スイッチで（**no bloggerd log-dump**が設定されていて）ログ ファイル保持機能が有効になっていない場合、次の手順を使用して単一のサービスに対して有効にします。

手順の概要

1. **show system internal sysmgr service name** *service-type*
2. **configure terminal**
3. **bloggerd log-dump sap** *number*
4. **show system internal bloggerd info log-dump-info**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show system internal sysmgr service name** *service-type* 例： `switch# show system internal sysmgr service name aclmgr`	サービス SA P番号を含む ACL Manager に関する情報を表示します。

ステップ 2	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 3	**bloggerd log-dump sap** *number* 例： `switch(config)# bloggerd log-dump sap 351`	ACL Managerサービスのログ ファイル保持機能をイネーブルにします。
ステップ 4	**show system internal bloggerd info log-dump-info** 例： `switch(config)# show system internal bloggerd info log-dump-info`	スイッチ上のログ ファイル保持機能に関する情報を表示します。

例

```
switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
        UUID = 0x182, PID = 653, SAP = 351
        State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
        Restart count: 1
        Time of last restart: Mon Nov  4 11:10:39 2019.
        The service never crashed since the last reboot.
        Tag = N/A
        Plugin ID: 0
switch(config)# configure terminal
switch(config)# bloggerd log-dump sap 351
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)# show system internal bloggerd info log-dump-info
------------------------------------------------------------
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
------------------------------------------------------------
Module     | VDC        | SAP                        | Enabled?
------------------------------------------------------------
1          | 1          | 351 (MTS_SAP_ACLMGR     ) | Enabled
------------------------------------------------------------
------------------------------------------------------------
Log Dump Throttle Switch-Wide Config:
-------------------------------------
Log Dump Throttle                                    : ENABLED
Minimum buffer rollover count (before throttling)    : 5
Maximum allowed rollover count per minute            : 1
------------------------------------------------------------

switch(config)#
```

拡張ログ ファイルの表示

スイッチに現在保存されているイベント ログ ファイルを表示するには、次の作業を実行します。

手順の概要

1. **dir debug:log-dump/**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**dir debug:log-dump/** 例： `switch# dir debug:log-dump/`	スイッチに現在保存されているイベント ログ ファイルを表示します。

例

```
switch# dir debug:log-dump/

3676160 Dec 05 02:43:01 2019 20191205023755_evtlog_archive.tar
3553280 Dec 05 06:05:06 2019 20191205060005_evtlog_archive.tar

Usage for debug://sup-local
913408 bytes used
4329472 bytes free
5242880 bytes total
```

ログ統計ごとのグローバル ディクショナリの表示

この CLI は、各コンポーネントによって記録されたログ メッセージの統計をカウンタとともに表示し、システムの稼働時間中に記録されたログの繰り返し回数を保存します。

手順の概要

1. **show system internal sdwrap buffers sap <sap-num> dict-stats detailed**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show system internal sdwrap buffers sap <sap-num> dict-stats detailed** 例： `switch# show system internal sdwrap buffers sap <sap-num> dict-stats detailed`	各コンポーネントのログごとの統計を表示します。

例

```
switch# show system internal sdwrap buffers sap 221 dict-stats detailed

Sap received is: 221

SDWrap Format Strings Dictionary stats for sap MTS_SAP_L2FM (221)

UUID: SRVUUID_LIBSDWRAP, Inst Type: 0

MsgId Frequency Message                                                    ---------
--------- ------------------------------
 4    1 System is not undergoing ISSU
78    1 Vlan %d is part of reserved vlan bmp from sdb                        179   1 Vlan
 %d is not found in L2FM database. Skipping the delete request 306   1 Vlan %d is removed
 from L2FM database and MTM database
416   1 mts_drap_get_my_local_swid_only_msg failed with rc %#x
496   1 Lookup for backplane mac failed for vdc %d with st = %s
598   1 L2FM - Slot %d SwCardId %d Port %d - %d Fp %d Cli %d
```

単一サービスに対する拡張ログファイル保持の無効化

拡張ログファイル保持は、スイッチ上のすべてのサービスに対してデフォルトで有効になっています。スイッチで単一またはすべてのサービス（Cisco NX-OSリリース9.3(5) ではデフォルト）に対してログファイル保持機能が有効になっている場合に、特定のサービスを無効にするには、次の手順を実行します。

手順の概要

1. **show system internal sysmgr service name** *service-type*
2. **configure terminal**
3. **no bloggerd log-dump sap** *number*
4. **show system internal bloggerd info log-dump-info**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show system internal sysmgr service name** *service-type* 例： `switch# show system internal sysmgr service name aclmgr`	サービス SA P番号を含む ACL Manager に関する情報を表示します。

ステップ 2	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 3	**no bloggerd log-dump sap** *number* 例： `switch(config)# no bloggerd log-dump sap 351`	ACL Managerサービスのログ ファイル保持機能を無効にします。
ステップ 4	**show system internal bloggerd info log-dump-info** 例： `switch(config)# show system internal bloggerd info log-dump-info`	スイッチ上のログ ファイル保持機能に関する情報を表示します。

例

次に、「aclmgr」という名前のサービスの拡張ログ ファイル保持を無効にする例を示します。

```
switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
        UUID = 0x182, PID = 653, SAP = 351
        State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
        Restart count: 1
        Time of last restart: Mon Nov  4 11:10:39 2019.
```

```
        The service never crashed since the last reboot.
        Tag = N/A
        Plugin ID: 0
switch(config)# configure terminal
switch(config)# no bloggerd log-dump sap 351
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)# show system internal bloggerd info log-dump-info
------------------------------------------------------------
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
------------------------------------------------------------
Module     | VDC        | SAP                        | Enabled?
------------------------------------------------------------
1          | 1          | 351 (MTS_SAP_ACLMGR     ) | Disabled
------------------------------------------------------------
------------------------------------------------------------
Log Dump Throttle Switch-Wide Config:
-------------------------------------
Log Dump Throttle                                    : ENABLED
Minimum buffer rollover count (before throttling)    : 5
Maximum allowed rollover count per minute            : 1
------------------------------------------------------------

switch(config)#
```

トリガーベースのイベント ログの自動収集

トリガーベースのログ収集機能：

- 問題発生時に関連データを自動的に収集します。
- コントロール プレーンへの影響なし
- カスタマイズ可能な設定ですか：
  - シスコが入力するデフォルト
  - 収集対象は、ネットワーク管理者または Cisco TACによって、選択的に上書きされます。
  - イメージのアップグレード時は新しいトリガーを自動的に更新します。
- ログをスイッチにローカルに保存するか、外部サーバにリモートで保存します。
- 重大度 0、1、および 2 の syslog をサポートします。
- BGP、BFD、OSPF、ISIS などの予期しないプロトコル イベントをサポートします。
- アドホック イベントのカスタム syslog（syslog と接続する自動収集コマンド）

トリガーベースのログ ファイルの自動収集の有効化

ログ ファイルのトリガーベースの自動作成を有効にするには、__syslog_trigger_default システム ポリシーのオーバーライド ポリシーをカスタム YAML ファイルで作成し、情報を収集する特定のログを定義する必要があります。

ログ ファイルの自動収集を有効にするカスタム YAML ファイルの作成の詳細については、自動収集 YAML ファイルの設定 （37 ページ） を参照してください。

ログプロファイル YAML ファイル

Log-Profile YAML ファイルは、コンポーネントのスロットル制限を定義するために使用されます。**log_profile.yaml** ファイルは、スイッチの /bootflash ディレクトリにあります。

Bloggerd は、コンポーネント名とロールオーバー情報を保持し、特定のコンポーネントのグローバル YAML ファイルで定義されている制限に基づいてログ ファイルを保存/保持します。

デフォルトでは、スイッチのスロットル値は 5 です。**log_profile.yaml** ファイルにエントリを追加すれば、スロットル カウントをオーバーライドできます。

`/bootflash/log_profile.yaml` ファイルに加えられた変更を反映するには、bloggerd の実行時に次の CLI を実行します。

- `switch# bloggerd reparse log-profile`

ログプロファイル YAML ファイルの例

以下は、イメージの一部としてパッケージ化されたデフォルトの log_profile.yaml ファイルの例です。ファイル内のキー/値の定義を次の表に示します。

```
273:
    entry_1:
          srv_uuid: 273
          instance: 0
          rollovers_allowed: 250
          rotations_allowed: 5
          mod: sup

274:
    entry_1:
          srv_uuid: 274
          instance: 0
          rollovers_allowed: 250
          rotations_allowed: 5
          mod: sup
```

キー：値	説明
273	sdwrap バッファー スロットリングをオーバーライドする必要があるコンポーネントの UUID。
entry_1：	コンポーネントごとにサポートされるエントリは 1 つだけです コンポーネントごとに最大20のエントリを作成できます。各エントリは、**entry_1** ～ **entry_20** として識別されます。

キー：値	説明
srv_uuid：	各 sdwrap ログ バッファーは、(uuid, instance ID) のタプルで識別されます。
instance:	上記の srv_uuid フィールドに関連する sdwrap ログ バッファ インスタンス ID。「**-1**」は、すべてのインスタンスを意味します。
rollovers_allowed：	1 分あたりに許可されるロールオーバーの数。許容値は **0 ～ 500** です。
回転の許可：	スロットルごとに許容される回転数。
mod：	syslog コンポーネントの名前（`platform` は syslog のファシリティ名）。

自動収集 YAML ファイル

EEM 機能の **action** コマンドで指定される自動収集 YAML ファイルは、さまざまなシステムまたは機能コンポーネントのアクションを定義します。このファイルは、スイッチ ディレクトリ: /bootflash/scripts にあります。デフォルトの YAML ファイルに加えて、コンポーネント固有の YAML ファイルを作成し、同じディレクトリに配置できます。コンポーネント固有の YAML ファイルの命名規則は **component-name.yaml** です。コンポーネント固有のファイルが同じディレクトリに存在する場合は、**action** コマンドで指定されたファイルよりも優先されます。たとえば、アクションファイル **bootflash/scripts/platform.yaml** がデフォルトのアクションファイル **/bootflash/scripts** とともに **bootflash/scripts/test.yaml** ディレクトリにある場合、**platform.yaml** ファイルで定義された命令がデフォルトの **test.yaml** ファイルに存在するプラットフォーム コンポーネントの手順よりも優先します。

コンポーネントの例としては、ARP、BGP、IS-IS などがあります。すべてのコンポーネント名に精通していない場合は、シスコ カスタマー サポートに連絡して、コンポーネント固有のアクション（およびデフォルトの **test.yaml** ファイル）の YAML ファイルを定義してください。

例：

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

コンポーネントごとの自動収集の作成または削除

Cisco NX-OS リリース 10.2(2)F 以降、自動収集の採用改善機能により、要件に基づいて単一または一連のコンポーネントの自動収集を制御できます。自動収集 YAML ファイルの作成または削除には、次のコマンドを使用できます。

（注） Cisco NX-OS リリース 10.3(1)F 以降、複数のコンポーネントがデフォルトで有効になり、コンポーネントの YAML ファイルがデフォルトの自動収集フォルダにコピーされます。ただし、このコマンドを使用して、ブロガーの自動収集コンポーネントを無効または有効にすることはできます。

```
switch# bloggerd auto-collect component <component_name> {enable | disable}
```

enable コマンドを使用すると、コンポーネントの yaml ファイルがバックアップ フォルダからデフォルトの自動収集フォルダにコピーされます。バックアップ ステージング フォルダは読み取り専用フォルダであるため、そこにはコンテンツをコピーできないことに注意してください。一方、必要に応じて、デフォルトの自動収集フォルダ（bootflash:scripts フォルダ）にはコンテンツをコピーできます。

disable コマンドを使用すると、コンポーネントの yaml ファイルが、bootflash:scripts フォルダの下のデフォルトの自動収集フォルダから削除されます。

出力例は次のようになります。

```
switch# bloggerd auto-collect component arp enable
Component arp auto-collect successfully enabled.
arp.yaml file copied from /bootflash/scripts/backup-staging to
/bootflash/scripts/default-autocollect
switch# dir bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml
Usage for bootflash://sup-local
11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
switch# dir bootflash:scripts/backup-staging/
switch# bloggerd auto-collect component ?
  CrdCfg         Auto-collect for CRDCFG
  aclmgr         Auto-collect for ACLMgr
  aclqos         Auto-collect for ACLQOS
  adjmgr         Auto-collect for Adjacency Manager
  arp            Auto-collect for ARP
  bcm_usd        Auto-collect for BCM USD
  bgp            Auto-collect for BGP
  cardclient     Auto-collect for CARD CLIENT
  cdp            Auto-collect for CPD
  cfs            Auto-collect for CFS
  clis           Auto-collect for CLIS
  cts            Auto-collect for CTS
  dhcp_snoop     Auto-collect for DHCP Snoop
  eigrp          Auto-collect for EIGRP
  eltm           Auto-collect for ELTM
  ethport        Auto-collect for Eth Port Manager
  feature-mgr    Auto-collect for Feature Manager
  fex            Auto-collect for Fex (Satellite Manager)
  hmm            Auto-collect for HMM
  hsrp_engine    Auto-collect for HSRP
  icam           Auto-collect for ICAM
  icmpv6         Auto-collect for ICMPv6
  iftmc          Auto-collect for IFTMC
  im             Auto-collect for IM
  ip             Auto-collect for IP
  ipfib          Auto-collect for IPFIB Manager
  ipqosmgr       Auto-collect for QOS Manager
  isis           Auto-collect for ISIS
  jer_usd        Auto-collect for JER USD
  kafka          Auto-collect for KAFKA Manager
  kern           Auto-collect for Kernel
  l2fm           Auto-collect for L2FM
  l2rib          Auto-collect for L2RIB
  l3vm           Auto-collect for L3VM
  lacp           Auto-collect for LACP
  lldp           Auto-collect for LLDP
  m2rib          Auto-collect for M2RIB
```

```
  mfdm           Auto-collect for MFDM
  mrib           Auto-collect for MRIB
  nbm            Auto-collect for NBM Daemon
  netstack       Auto-collect for Netstack
  ngoam          Auto-collect for NGOAM
  nve            Auto-collect for NVE
  ospf           Auto-collect for Open Shortest Path First Unicast Routing Protocol
(OSPF)
  ospfv3         Auto-collect for Open Shortest Path First Version 3 Unicast Routing
Protocol
  pfma           Auto-collect for PFM
  pim            Auto-collect for PIM
  pktmgr         Auto-collect for Packet Manager
  pltfm_config   Auto-collect for PLTFM CONFIG
  port-channel   Auto-collect for Port Channel Manager
  qos            Auto-collect for QOS Manager
  rip            Auto-collect for RIP
  sdaa           Auto-collect for SDAA
  sla_responder  Auto-collect for SLA Responder
  sla_sender     Auto-collect for SLA Sender
  sla_twamp      Auto-collect for SLA Twamp
  smm            Auto-collect for SMM
  snmpmib_proc   Auto-collect for Snmpmib_proc
  spm            Auto-collect for SPM
  statsclient    Auto-collect for Statistics Client
  sysmgr         Auto-collect for SYSMGR
  tahusd         Auto-collect for TAHUSD
  tctrl_usd      Auto-collect for TCTRL USD
  tun_enc_mgr    Auto-collect for TEM
  udld           Auto-collect for UDLD
  ufdm           Auto-collect for UFDM
  vmtracker      Auto-collect for VMTRACKER
  vntag_mgr      Auto-collect for VNTAG Mgr
  vpc            Auto-collect for VPC
  vrrp-cfg       Auto-collect for VRRP Configuration
  vrrp-eng       Auto-collect for VRRP Engine
  vrrpv3         Auto-collect for VRRPV3
Usage for bootflash://sup-local 11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
switch# dir bootflash:scripts/default-autocollect^C n9k-A# dir
bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml Usage for bootflash://sup-local 11078049792 bytes
used
10653151232 bytes free
21731201024 bytes total
```

以下は、UDLD コンポーネント用に事前入力された YAML ファイルを作成する例です。

```
n9k-A# bloggerd auto-collect component udld enable
Component udld auto-collect successfully enabled.
udld.yaml file copied from /bootflash/scripts/backup-staging to
/bootflash/scripts/default-autocollect
n9k-A# dir bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml
431 Nov 10 08:44:45 2021 udld.yaml
Usage for bootflash://sup-local
11078053888 bytes used
10653147136 bytes free
21731201024 bytes total
n9k-A# sh running-config all | include bloggerd
```

```
bloggerd log-dump all
bloggerd log-throttle
no bloggerd log-transfer
```

以下は、UDLD コンポーネント用に事前入力された YAML ファイルを削除する例です。

```
n9k-A# bloggerd auto-collect component udld disable
Component udld auto-collect successfully disabled.
udld.yaml file deleted from /bootflash/scripts/default-autocollect
n9k-A# dir bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml
Usage for bootflash://sup-local
11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
n9k-A#
```

プロトコル フラップの自動収集の作成

Cisco NX-OS リリース 10.4(1)F 以降では、予期しないプロトコル フラップ イベントが発生した場合、BGP、OSPF、ISIS、および BFD プロトコルの自動データ収集がデフォルトで有効になります。

既存の自動収集は、重大度 0、1、および 2 の syslog メッセージに対して機能します。予期しないプロトコルフラップ自動収集は、ルーティング プロトコルの予期しないイベントが特定のプロトコル プロセスによって検出されるとトリガされます。この自動収集は、制御システム EEM ポリシー `trigger_generic_evt_default` です。

次に、BGP コンポーネントでデフォルトの自動収集を有効にする例を示します。

```
switch# configure terminal
switch(config)# bloggerd auto-collect component bgp enable
component bgp auto-collect successfully enabled
bgp.yaml file copied from /bootflash/scripts/backup-staging to
/bootflash/scripts/default-autocollect
switch(config)# run bash
bash-4.4$ cd /bootflash/scripts/default-autocollect/
bash-4.4$ ls bgp.yaml
```

次に、スイッチでの自動収集履歴の例を示します。

```
switch(config)# show system internal event-logs auto-collect history
DateTime    SnapshotID  Syslog      Status/Secs/Logsize(Bytes)
2023-Mar-21 11:04:03 1395380375    NVE-0- TEST_SYSLOGPROCESSED:23:5756541
```

自動収集 YAML ファイルの設定

YAML ファイルの内容によって、トリガーベースの自動収集時に収集されるデータが決まります。スイッチには YAML ファイルが 1 つだけ存在しますが、任意の数のスイッチ コンポーネントとメッセージの自動収集メタデータを含めることができます。

スイッチの次のディレクトリで YAML ファイルを見つけます。

```
/bootflash/scripts
```

次の例を使用して、トリガーベース収集の YAMLファイルを呼び出します。この例は、ユーザ定義の YAML ファイルを使用してトリガーベース収集を実行するために最低限必要な設定を示しています。

```
switch# show running-config eem
!Command: show running-config eem
!Running configuration last done at: Mon Sep 30 19:34:54 2019
!Time: Mon Sep 30 22:24:55 2019
version 9.3(3) Bios:version 07.59
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

上記の例では、「test_1」がアプレットの名前で、「test.yaml」が /bootflash/scripts ディレクトリにあるユーザ設定の YAML ファイルの名前です。

YAML ファイルの例

次に、トリガーベースのイベント ログ自動収集機能をサポートする基本的な YAML ファイルの例を示します。ファイル内のキー/値の定義を次の表に示します。

（注） YAML ファイルに適切なインデントがあることを確認します。ベスト プラクティスとして、スイッチで使用する前に任意の「オンライン YAML 検証」を実行します。

```
bash-4.3$ cat /bootflash/scripts/test.yaml
version: 1
components:
   securityd:
        default:
            tech-sup: port
            commands: show module
   platform:
        default:
            tech-sup: port
            commands: show module
```

キー：値	説明
バージョン：1	1 に設定します。他の番号を使用すると、自動収集スクリプトに互換性がなくなります。
コンポーネント：	以下がスイッチ コンポーネントであることを指定するキーワード。
securityd：	syslog コンポーネントの名前（securityd は syslog のファシリティ名）。
デフォルト：	コンポーネントに属するすべてのメッセージを識別します。
tech-sup：port	securityd syslog コンポーネントのポート モジュールのテクニカル サポートを収集します。
コマンド：show module	securityd syslog コンポーネントの show module コマンド出力を収集します。
プラットフォーム：	syslog コンポーネントの名前（platform は syslog のファシリティ名）。

キー：値	説明
tech-sup：port	platform syslog コンポーネントのポート モジュールのテクニカル サポートを収集します。
コマンド：show module	platform syslog コンポーネントの show module コマンド出力を収集します。

特定のログにのみ自動収集メタデータを関連付けるには、次の例を使用します。たとえば、SECURITYD-2-FEATURE_ENABLE_DISABLE

```
securityd:
                feature_enable_disable:
                        tech-sup: security
                        commands: show module
```

キー：値	説明
securityd：	syslog コンポーネントの名前（securityd は syslog のファシリティ名）。
feature_enable_disable：	syslog メッセージのメッセージ ID。
tech-sup：security	securityd syslog コンポーネントのセキュリティモ ジュールのテクニカル サポートを収集します。
コマンド：show module	セキュリティ syslog コンポーネントの show module コマンド出力を収集します。

上記の YAML エントリの syslog 出力の例：

```
2019 Dec 4 12:41:01 n9k-c93108tc-fx %SECURITYD-2-FEATURE_ENABLE_DISABLE: User
has enabled the feature bash-shell
```

複数の値を指定するには、次の例を使用します。

```
version: 1
components:
   securityd:
        default:
            commands: show module;show version;show module
            tech-sup: port;lldp
```

（注） 複数の show コマンドとテクニカル サポート キーの値を区切るには、セミコロンを使用します（前の例を参照）。

リリース 10.1(1) 以降では、test.yaml は複数の YAML ファイルが存在するフォルダに置き換えることができます。フォルダ内のすべての YAML ファイルは、ComponentName.yaml 命名規則に従う必要があります。

次の例では、test.yaml が test_folder に置き換えられます。

```
test.yaml:
event manager applet logging2 override __syslog_trigger_default
   action 1.0 collect test.yaml rate-limt 30 $_syslog_msg

test_folder:
event manager applet logging2 override __syslog_trigger_default
   action 1.0 collect test_folder rate-limt 30 $_syslog_msg
```

次の例は、`test_folder` のパスとコンポーネントを示しています。

```
ls /bootflash/scripts/test_folder
bgp.yaml ppm.yaml
```

コンポーネントあたりの自動収集の量の制限

自動収集の場合、コンポーネントイベントあたりのバンドル数の制限は、Cisco NX-OS リリース 10.2(2)F 以降では、デフォルトで 1 に設定されています。それより前は、この制限はデフォルトで 3 です。1 つのコンポーネントでデフォルトよりも多くのイベントが発生すると、イベントはドロップされ、ステータス メッセージ **EVENTLOGLIMITREACHED** が表示されます。イベント ログがロールオーバーすると、コンポーネント イベントの自動収集が再開されます。

例：

```
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog                   Status/Secs/Logsize(Bytes)
2020-Jun-27 07:20:03  1140276903   ACLMGR-0-TEST_SYSLOG     EVENTLOGLIMITREACHED
2020-Jun-27 07:15:14  1026359228   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:15:09  384952880    ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:13:55  1679333688   ACLMGR-0-TEST_SYSLOG     PROCESSED:2:9332278
2020-Jun-27 07:13:52  1679333688   ACLMGR-0-TEST_SYSLOG     PROCESSING
2020-Jun-27 07:12:55  502545693    ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:12:25  1718497217   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:08:25  1432687513   ACLMGR-0-TEST_SYSLOG     PROCESSED:2:10453823
2020-Jun-27 07:08:22  1432687513   ACLMGR-0-TEST_SYSLOG     PROCESSING
2020-Jun-27 07:06:16  90042807     ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:03:26  1737578642   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:02:56  40101277     ACLMGR-0-TEST_SYSLOG     PROCESSED:3:10542045
2020-Jun-27 07:02:52  40101277     ACLMGR-0-TEST_SYSLOG     PROCESSING
```

自動収集ログ ファイル

自動収集ログ ファイルについて

YAML ファイルの設定によって、自動収集ログ ファイルの内容が決まります。収集ログ ファイルで使用されるメモリの量は設定できません。保存後のファイルが消去される頻度は設定できます。

自動収集ログ ファイルは、次のディレクトリに保存されます。

```
switch# dir bootflash:eem_snapshots
   44205843    Sep 25 11:08:04 2019
1480625546_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz
  Usage for bootflash://sup-local
 6940545024 bytes used
44829761536 bytes free
51770306560 bytes total
```

ログ ファイルへのアクセス

コマンド キーワード「debug」を使用してログを検索します。

```
switch# dir debug:///
...
       26    Oct 22 10:46:31 2019  log-dump
       24    Oct 22 10:46:31 2019  log-snapshot-auto
       26    Oct 22 10:46:31 2019  log-snapshot-user
```

次の表に、ログの場所と保存されるログの種類を示します。

場所	説明
log-dump	このフォルダには、ログ ロールオーバー時にイベント ログが保存されます。
log-snapshot-auto	このフォルダには、syslogイベント0、1、2の自動収集ログが含まれます。
log-snapshot-user	このフォルダには、bloggerd log-snapshot の実行時に収集されたログが保存されます。

ログ ロールオーバーで生成されたログ ファイルを表示するには、次の例を参考にしてください。

```
switch# dir debug:log-dump/
debug:log-dump/20191022104656_evtlog_archive.tar
debug:log-dump/20191022111241_evtlog_archive.tar
debug:log-dump/20191022111841_evtlog_archive.tar
debug:log-dump/20191022112431_evtlog_archive.tar
debug:log-dump/20191022113042_evtlog_archive.tar
debug:log-dump/20191022113603_evtlog_archive.tar
```

ログ tar ファイルの解析

tar ファイル内のログを解析するには、次の例を参考にしてください。

```
switch# show system internal event-logs parse
debug:log-dump/20191022104656_evtlog_archive.tar
--------LOGS:/tmp/BLOGGERD0.991453012199/tmp/1-191022104658-191022110741-device_test-M27-V1-I1:0-P884.gz--------
2019 Oct 22 11:07:41.597864 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Data Space
Limits(bytes): Soft: -1  Ha rd: -1
2019 Oct 22 11:07:41.597857 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Stack Space
Limits(bytes): Soft: 500000  Hard: 500000
2019 Oct 22 11:07:41.597850 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):AS: 1005952076
 -1
2019 Oct 22 11:07:41.597406 E_DEBUG Oct 22 11:07:41 2019(device_test_process_events):Sdwrap
 msg unknown
2019 Oct 22 11:07:41.597398 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Going back to
 select
2019 Oct 22 11:07:41.597395 E_DEBUG Oct 22 11:07:41 2019(nvram_test):TestNvram examine
27 blocks
2019 Oct 22 11:07:41.597371 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Parent: Thread
 created test index:4 thread_id:-707265728
2019 Oct 22 11:07:41.597333 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Node inserted
2019 Oct 22 11:07:41.597328 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):The test index
 in diag is 4
2019 Oct 22 11:07:41.597322 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):result severity
 level
2019 Oct 22 11:07:41.597316 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):callhome alert
 level
```

次の表に、特定の tar ファイルの解析に使用できる追加のキーワードを示します。

キーワード	説明
component	プロセス名で識別されるコンポーネントに属するログをデコードします。
from-datetime	yy [mm [dd [HH [MM [SS]]]]] 形式で指定した、特定の日時のログをデコードします。
instance	デコードする SDWRAP バッファ インスタンスのリスト（カンマ区切り）。
module	SUP や LC などのモジュールからのログをデコードします（モジュール ID を使用）。
to-datetime	yy [mm [dd [HH [MM [SS]]]]] 形式で指定した、特定の日時までのログをデコードします。

別の場所へログをコピーする

リモート サーバなどの別の場所にログをコピーするには、次の例を参考にしてください。

```
switch# copy debug:log-dump/20191022104656_evtlog_archive.tar
scp://<ip-adress>/nobackup/<user> vrf management  use-kstack
Enter username: user@<ip-address>'s password:
20191022104656_evtlog_archive.tar                                        100%  130KB
 130.0KB/s   00:00
Copy complete, now saving to disk (please wait)...
Copy complete.
```

自動収集ログファイルの消去

生成されるトリガー ベースの自動収集ログには、EventHistory と EventBundle の 2 種類があります。

EventHistory ログの消去ロジック

イベント履歴の場合は、/var/sysmgr/srv_logs/xport フォルダで消去が行われます。250 MB のパーティション RAM が、/var/sysmgr/srv_logs ディレクトリにマウントされます。

/var/sysmgr/srv_logs のメモリ使用率が、割り当てられた 250 MB の 65% 未満の場合、ファイルは消去されません。メモリ使用率が 65% の制限レベルに達すると、新しいログの保存を続行するのに十分なメモリが使用可能になるまで、最も古いファイルから消去されます。

EventBundle ログの消去ロジック

イベント バンドルの場合、消去ロジックは /bootflash/eem_snapshots フォルダでの状態に基づいて実行されます。自動収集されたスナップショットを保存するために、EEM 自動収集スクリプトは、ブートフラッシュ ストレージの 5% を割り当てます。ブート フラッシュ容量の 5% が使用されると、ログは消去されます。

新しい自動収集ログが利用可能になっているものの、ブートフラッシュに保存するスペースがない場合（すでに 5% の容量に達している）、システムは次のことを確認します。

1. 12時間以上経過した既存の自動収集ファイルがある場合、システムはファイルを削除し、新しいログをコピーします。
2. 既存の自動収集ファイルが 12 時間未満の場合、新しく収集されたログは保存されずに廃棄されます。

デフォルトパージ時間である 12 時間は、次のコマンドを使用して変更できます。コマンドで指定する時間は分単位です。

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml purge-time 300 $_syslog_msg
```

event manager command: *test* は、ポリシー例の名前です。**__syslog_trigger_default** は、オーバーライドする必要のあるシステム ポリシーの名前です。この名前は、二重アンダースコア（__）で始まる必要があります。

action command: **1.0** は、アクションの実行順番を示している例となっています。**collect** は、データが YAMU ファイルを使用して収集されることを示しています。*test.yaml* は、YAML ファイルの名前の例です。**$_syslog_msg** は、コンポーネントの名前です。

（注） どの時点でも、進行中のトリガーベースの自動収集イベントは1つだけです。自動収集がすでに発生しているときに別の新しいログ イベントを保存しようとすると、新しいログ イベントは破棄されます。

デフォルトでは、トリガーベースのバンドルは 5 分（300 秒）ごとに 1 つだけ収集されます。このレート制限は、次のコマンドでも設定できます。コマンドで指定する時間は秒単位です。

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml rate-limit 600 $_syslog_msg
```

event manager command: *test* はポリシーの名前の例です。**__syslog_trigger_default** は、オーバーライドするシステム ポリシーの名前の例です。この名前は、二重アンダースコア（__）で始まる必要があります。

action command: **1.0** は、アクションの実行順番を示している例となっています。**collect** は、データが YAMU ファイルを使用して収集されることを示しています。*test.yaml* は、YAML ファイルの名前の例です。**$_syslog_msg** は、コンポーネントの名前です。

リリース 10.1(1) 以降では、トリガーの最大数オプションを使用して収集レートを調整することもできます。これは、この数のトリガーだけを保つものです。 **max-triggers** の値に達すると、syslog が発生しても、これ以上バンドルは収集されなくなります。

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml rate-limt 30 max-triggers 5 $_syslog_msg
```

（注） 自動収集されたバンドルを `debug:log-snapshot-auto/` により手動で削除すれば、次のイベントが発生したとき、 **max-triggers** の設定数に基づいて収集が再開されます。

自動収集の統計情報と履歴

トリガーベースの収集統計情報の例を次に示します。

```
switch# show system internal event-logs auto-collect statistics
--------------------EEM Auto Collection Statistics--------------------
Syslog Parse Successful :88 Syslog Parse Failure :0
Syslog Ratelimited :0 Rate Limit Check Failed :0
Syslog Dropped(Last Action In Prog) :53 Storage Limit Reached :0
User Yaml Action File Unavailable :0 User Yaml Parse Successful :35
User Yaml Parse Error :0 Sys Yaml Action File Unavailable :11
Sys Yaml Parse Successful :3 Sys Yaml Parse Error :0
Yaml Action Not Defined :0 Syslog Processing Initiated :24
Log Collection Failed :0 Tar Creation Error :0
Signal Interrupt :0 Script Exception :0
Syslog Processed Successfully :24 Logfiles Purged :0
```

次の例は、CLI コマンドを使用して取得されたトリガーベースの収集履歴（処理された syslog 数、処理時間、収集されたデータのサイズ）を示しています。

```
switch# show system internal event-logs auto-collect history
DateTime Snapshot ID Syslog Status/Secs/Logsize(Bytes)
2019-Dec-04 05:30:32 1310232084 VPC-0-TEST_SYSLOG PROCESSED:9:22312929
2019-Dec-04 05:30:22 1310232084 VPC-0-TEST_SYSLOG PROCESSING
2019-Dec-04 04:30:13 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSED:173:33194665
2019-Dec-04 04:28:47 897805674 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:28:47 947981421 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:27:19 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSING
2019-Dec-04 02:17:16 1957148102 CARDCLIENT-2-FPGA_BOOT_GOLDEN NOYAMLFILEFOUND
```

サポートされているログ収集の例

いくつかのコンポーネントでサポートされるログのサンプル コレクションを次に示します。

コンポーネント：**IPQoSMgr**

サポートされるログ：

```
QOSMGR_MTS_FAILURE
QOSMGR_NETWORK_QOS_POLICY_CHANGE
QOSMGR_LLFC_APPLY_FAILURE
QOSMGR_FCOE_POLICY_NOT_REMOVED
```

コンポーネント：**ACLQOS**

サポートされるログ：

```
ACLQOS_UNEXPECTED_MCAST_FRAMES
ACLQOS_UNEXPECTED_PFC_FRAMES
PPF_SUBSCRIPTION_FAILED
ACLQOS_QOS_NO_DROP_CLASSIFICATION_UNSUPPORTED
ACLQOS_QUEUE_LIMIT_IGNORED_ON_FEX
ACLQOS_BUFFER_DRAIN_FAILURE
ACLQOS_BURST_DETECT_FPGA_INCOMPATIBLE
ACLQOS_BURST_DETECT_OVER_THRESHOLD
ACLQOS_FAILED
PPF_FAILED
```

トリガーベースのログ収集の確認

次の例のように **show event manager system-policy | i trigger** コマンドを入力して、トリガーベースのログ収集機能が有効になっていることを確認します。

```
switch# show event manager system-policy | i trigger n 2
          Name : __syslog_trigger_default
    Description : Default policy for trigger based logging
    Overridable : Yes
     Event type : 0x2101
```

トリガーベースのログ ファイル生成の確認

トリガーベースの自動収集機能によってイベント ログ ファイルが生成されたかどうかを確認できます。次の例のいずれかのコマンドを入力します。

```
switch# dir bootflash:eem_snapshots
9162547 Nov 12 22:33:15 2019
1006309316_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz

Usage for bootflash://sup-local
8911929344 bytes used
3555950592 bytes free
12467879936 bytes total
```

```
switch# dir debug:log-snapshot-auto/
63435992 Dec 03 06:28:52 2019
20191203062841_1394408030_PLATFORM_2_MOD_PWRDN_eem_snapshot.tar.gz

Usage for debug://sup-local
544768 bytes used
4698112 bytes free
5242880 bytes total
```

ローカル ログ ファイルのストレージ

ローカル ログ ファイルのストレージ機能：

- ローカルデータ ストレージ時間の量は、導入の規模とタイプによって異なります。モジュラ スイッチと非モジュラ スイッチの両方で、ストレージ時間は 15 分から数時間のデータです。長期間にわたる関連ログを収集するには、次の手順を実行します。
  - 必要な特定のサービス/機能に対してのみイベント ログの保持を有効にします。「単一サービスの拡張ログファイル保持の有効化 （28 ページ）」を参照してください。
  - スイッチから内部イベント ログをエクスポートします。「外部ログ ファイルのストレージ （48 ページ）」を参照してください。
- 圧縮されたログは RAM に保存されます。
- 250MB のメモリは、ログ ファイル ストレージ用に予約されています。
- ログ ファイルは tar 形式で最適化されます（5 分ごとに 1 ファイルまたは 10 MB のいずれか早い方）。
- スナップ ショット収集を許可します。

最近のログ ファイルのローカル コピーの生成

拡張ログ ファイル保持は、スイッチで実行されているすべてのサービスに対してデフォルトで有効になっています。ログ ファイルは、フラッシュ メモリにローカルに保存されます。次の手順を使用して、最新のイベント ログ ファイルを最大 10 個生成します。

手順の概要

1. **bloggerd log-snapshot** [*file-name*] [**bootflash:** *file-path* | **logflash:** *file-path* | **usb1:**] [**size** *file-size*] [**time** *minutes*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**bloggerd log-snapshot** [*file-name*] [**bootflash:** *file-path* | **logflash:** *file-path* | **usb1:**] [**size** *file-size*] [**time** *minutes*] 例： `switch# bloggerd log-snapshot snapshot1`	スイッチに保存されている最新の 10 個のイベント ログのスナップショット バンドル ファイルを作成します。この操作のデフォルトのストレージは **logflash** です。 *file-name*：生成されたスナップショット ログ ファイル バンドルのファイル名。*file-name* には最大 64 文字を使用します。 （注） この変数はオプションです。設定されていない場合、システムはタイムスタンプと「_snapshot_bundle.tar」をファイル名として適用します。例： `20200605161704_snapshot_bundle.tar` **bootflash:** *file-path*：スナップショット ログ ファイル バンドルがブート フラッシュに保存されているファイル パス。次の初期パスのいずれかを選択します。 • bootflash:/// • bootflash://module-1/ • bootflash://sup-1/ • bootflash://sup-active/ • bootflash://sup-local/ **logflash:** *file-path*：スナップショット ログ ファイル バンドルがログ フラッシュに保存されるファイル パス。次の初期パスのいずれかを選択します。 • logflash:/// • logflash://module-1/ • logflash://sup-1/

	コマンドまたはアクション	目的
		• logflash://sup-active/ • logflash://sup-local/ **usb1:**：USB デバイス上のスナップショット ログ ファイル バンドルが保存されているファイル パス。 **size** *file-size*：メガバイト（MB）単位のサイズに基づくスナップショット ログ ファイル バンドル。範囲は 5MB ー 250MB です。 **time** *minutes*：最後の x 時間（分）に基づくスナップショット ログ ファイル バンドル。範囲は 1 ～ 30 分です。

例

```
switch# bloggerd log-snapshot snapshot1
Snapshot generated at logflash:evt_log_snapshot/snapshot1_snapshot_bundle.tar Please
cleanup once done.
switch#
switch# dir logflash:evt_log_snapshot
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for logflash://sup-local
759865344 bytes used
5697142784 bytes free
6457008128 bytes total
```

次の例のコマンドを使用して、同じファイルを表示します。

```
switch# dir debug:log-snapshot-user/
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for debug://sup-local
929792 bytes used
4313088 bytes free
5242880 bytes total
```

（注） 例の最後のファイル名に注意してください。個々のログ ファイルは、生成された日時によっても識別されます。

リリース 10.1(1) 以降、LC コアファイルには `log-snapshot` バンドルが含まれています。`log-snapshot` バンドル ファイル名は、`tac_snapshot_bundle.tar.gz` です。次に例を示します。

```
bash-4.2$ tar -tvf 1610003655_0x102_aclqos_log.17194.tar.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 pss/
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_runtime_info_lc.gz
```

```
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_runtime_cfg_lc.gz
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_debug.gz
-rw-rw-rw- root/root 129583 2021-01-07 12:44 pss/clqosdb_ver1_0_user.gz
-rw-rw-rw- root/root 20291 2021-01-07 12:44 pss/clqosdb_ver1_0_node.gz
-rw-rw-rw- root/root 444 2021-01-07 12:44 pss/clqosdb_ver1_0_ctrl.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 proc/
-rw-rw-rw- root/root 15159 2021-01-07 12:44 0x102_aclqos_compress.17194.log.25162
-rw-rw-rw- root/root 9172392 2021-01-07 12:43 0x102_aclqos_core.17194.gz
-rw-rw-rw- root/root 43878 2021-01-07 12:44 0x102_aclqos_df_dmesg.17194.log.gz
-rw-rw-rw- root/root 93 2021-01-07 12:44 0x102_aclqos_log.17194
-rw-rw-rw- root/root 158 2021-01-07 12:44 0x102_aclqos_mcore.17194.log.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 usd17194/
-rw-rw-rw- root/root 11374171 2021-01-07 12:44 tac_snapshot_bundle.tar.gz
```

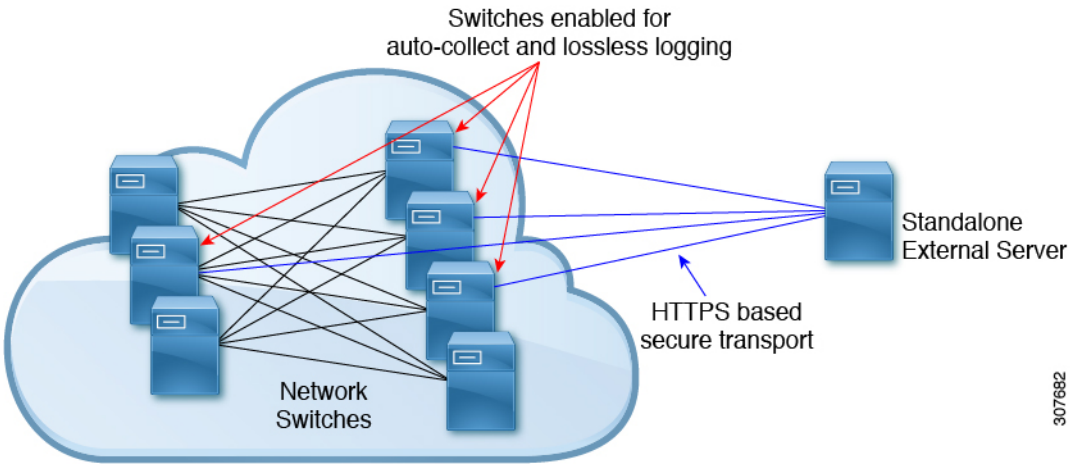
外部ログ ファイルのストレージ

外部サーバ ソリューションは、ログを安全な方法でオフスイッチに保存する機能を提供します。

（注） 外部ストレージ機能を作成するため、Cisco Technical Assistance Center（TAC）に連絡して、外部サーバ ソリューションの展開をサポートを求めてください。

次に、外部ログ ファイルの保存機能を示します。

- オンデマンドで有効
- HTTPS ベースの転送
- ストレージ要件：
  - 非モジュラ スイッチ：300 MB
  - モジュラ スイッチ：12 GB（1 日あたり、スイッチあたり）
- 通常、外部サーバには 10 台のスイッチのログが保存されます。ただし、外部サーバでサポートされるスイッチの数に厳密な制限はありません。



外部サーバ ソリューションには、次の特性があります。

- コントローラレス環境
- セキュリティ証明書の手動管理
- サポートされている 3 つの使用例：
  - 選択したスイッチからのログの継続的な収集
  - TAC のサポートによる、シスコ サーバへのログの展開とアップロード。
  - 限定的なオンプレミス処理

（注） 外部サーバでのログ ファイルの設定と収集については、Cisco TAC にお問い合わせください。

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。