ソフトウェア管理の運用

この章では、システム上のソフトウェア管理の運用について説明します。

ローカルファイルシステムの概要

VPC VM 上のローカルファイルシステムは、次のものに保存されているファイルで構成されます。

- **/flash** ハイパーバイザを介して M 上の vHDD-1 として割り当てられたフラッシュメモリは、システムで使用される StarOS イメージ、CLI 設定、およびクラッシュログファイルのデフォルトのストレージメディアです。
- **/hd-raid** これは、ハイパーバイザによって CF VM 上で vHDD-2 として割り当てられたストレージ領域です。CDR（課金データレコード）と UDR（使用データレコード）を保存するために使用されます。

ローカルファイルシステムで使用されるファイルタイプ

ローカルファイルシステムには、次のファイルタイプを配置できます。

- **オペレーティング システム ソフトウェアのイメージファイル**：このバイナリファイルタイプは **.bin** 拡張子で識別されます。ファイルは、起動時またはリロード時にシステムによってロードされるオペレーティングシステムです。これは、エンドユーザーが変更できない実行可能な読み取り専用ファイルです。

- **CLI 設定ファイル**：このファイルタイプは **.cfg** 拡張子によって識別されます。これらは、オペレーティングシステムのソフトウェアイメージと連携して動作する CLI コマンドを含むテキストファイルです。これらのファイルによって、提供されるサービス、ハードウェアとソフトウェアの設定、システムによって実行されるその他の機能が決まります。通常、ファイルはエンドユーザーが作成します。ファイルはオンラインでも、オフラインでも変更でき、そのファイルを説明する長いファイル名を使用します。
- **システムファイル**：システムでは、**.sys** 拡張子によって識別されるファイル 1 つのみが使用されます。boot.sys ファイルには、システムの検出方法や、起動スタックからファイルグループ（.bin ファイルと .cfg ファイルのペア）をロードする優先順位を説明するシステム固有の情報が含まれています。
- **クラッシュログの要約**：ファイル名の **crashlog** で識別されるクラッシュログの要約には、システムで発生したソフトウェアまたはハードウェアの障害に関するサマリー情報が含まれています。このファイルは、デバイスの **/flash/crsh2/** ディレクトリにあります。CLI を使用してこのファイルの内容を表示することはできますが、ファイルを変更することはできません。

boot.sys ファイルの概要

システムは boot.sys ファイルを使用して、システムが起動時に使用する優先順位付けされた起動スタックパラメータとファイルグループを保存します。このファイルは、外部手段ではなく、システム CLI コマンドを使用してのみ変更できます。起動パラメータには、オペレーティングシステムのイメージファイルを見つけるために必要な次のような情報が含まれています。

- **bootmode**：この設定は通常は normal に設定され、システムの起動方法を識別します。
- **起動スタック情報**：起動スタックは、オペレーティングシステムのイメージファイルと、ロードする CLI 設定ファイルを指定する、優先順位付けられたファイルグループエントリから構成されます。

システムを初めて起動したときに、通常の起動モードを使用して、/flash ディレクトリからオペレーティングシステムのソフトウェアイメージをロードするように、boot.sys ファイルが設定されます。

ローカルファイルシステムには CLI 設定ファイルが含まれていません。これにより、システムは最初に正常に起動したときに自動的に CLI ベースのクイック セットアップ ウィザードを開始します。クイック セットアップ ウィザードの使用方法の詳細については、「使用する前に」を参照してください。

ローカルファイルシステムの保守

ローカルファイルシステムを構成するデバイスを管理および保守するには、CLI コマンドを使用します。この項で説明されているすべてのコマンドは Exec モードで実行します。特に指定がない限り、これらのコマンドを実行するには、セキュリティ管理者または管理者の権限が必要です。

ファイルシステム管理コマンド

ローカルファイルシステムを管理および構成するには、この項のコマンドを使用します。

重要 次に示すコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

ディレクトリの作成

特定のローカルデバイスに新しいディレクトリを作成するには、**mkdir** コマンドを使用します。このディレクトリは、ローカルファイルシステムにある任意のファイルのパス名の一部として組み込むことができます。

```
[local]host_name# mkdir { /flash | /usb1 | /hd-raid } /dir_name
```

次のコマンドを使用して、*configs* という名前のディレクトリを作成します。

```
[local]host_name# mkdir /flash/configs
```

ファイルとディレクトリの名前の変更

ファイルの名前を元の名前から別の名前に変更するには、**rename** コマンドを使用します。必要に応じて同じファイル拡張子を使用し、ファイルタイプが変更されないようにします。

/flash ローカルデバイス上でファイル名 *iot_test.cfg* を *iot_accept.cfg* に変更するには、次のコマンドを使用します。

```
[local]host_name# rename /flash/iot_test.cfg /flash/iot_accept.cfg
-noconfirm
```

重要 **rename** コマンドは、同じローカルデバイス内でのみ使用します。ファイル名を変更して、そのファイルを別のローカルデバイスに同時に配置することはできません。名前を変更したファイルを移動するには、**copy** コマンドを使用する必要があります。

ファイルのコピー

これらの手順は、Execモードのルートプロンプトを使用していることが前提になります。現在の設定を保存するには、次のコマンドを入力します。

```
[local]host_name# copy from_url to_url [-noconfirm]
```

system.cfg という設定ファイルを *cfgfiles* と呼ばれていたディレクトリから *configs_old* というディレクトリにコピーするには、次のコマンドを入力します。

```
[local]host_name# copy /flash/cfgfiles/system.cfg
/flash/configs_old/system_2011.cfg
```

init_config.cfg という設定ファイルをホスト名が *config_server* の TFTP サーバーのルートディレクトリにコピーするには、次のコマンドを入力します。

```
[local]host_name# copy /flash/cfgfiles/init_confg.cfg
tftp://config_server/init_config.cfg
```

ファイルの削除

delete コマンドは、指定されたファイルをローカルファイルシステム上の指定された場所から削除します。

重要 このコマンドは、ワイルドカードエントリをサポートしていません。各ファイル名は、全体で指定する必要があります。

注意 boot.sys ファイルは削除しないでください。削除すると、システムはコマンドを再起動せず、動作不能になります。

```
[local]host_name# delete { /flash | /usb1 | /hd-raid }/filename [ -noconfirm
 ]
```

次のコマンドは、*/flash* ディレクトリから *est.cfg* という名前のファイルを削除します。

```
[local]host_name# delete /flash/test.cfg
```

ディレクトリの削除

rmdir コマンドは、特定のローカルデバイス上の現在のディレクトリを削除します。このディレクトリは、ローカルファイルシステムにある任意のファイルのパス名の一部として組み込むことができます。

重要 削除するディレクトリは、**rmdir** コマンドを実行する前に空にしておく必要があります。ディレクトリが空でない場合、CLI には「Directory not empty」というメッセージが表示され、実行されません。

```
[local]host_name# rmdir url /dir_name
```

次に、*/flash* ディレクトリ内の *config* という名前の空のディレクトリを削除するコマンドを示します。

```
[local]host_name# rmdir /flash/configs
```

ローカルデバイスのフォーマット

format コマンドは、ローカルデバイスの低レベルフォーマットを実行します。この操作は、FAT16 フォーマット方式を使用するようにデバイスをフォーマットします。これは、オペレーティングシステムの適切な読み取り/書き込み機能に必要です。

重要 NTFS や FAT32 などの他の方式を使用してフォーマットされたローカルデバイスは、さまざまなオペレーティングシステム、CLI 設定、およびクラッシュログファイルを保存するために使用される場合があります。ただし、通常使用のために新しいローカルデバイスを MIO/UMIO/MIO2 に配置する場合は、使用する前にシステムを介してデバイスをフォーマットする必要があります。これにより、適切なファイル アロケーション テーブルのフォーマットが使用され、他のオペレーティングシステムで使用されている他のフォーマットとの不一致を防止することができます。

注意 **filesystem format** コマンドは、デバイスに保存されているすべてのファイルと情報を削除します。

ローカルファイルシステムで使用するローカルデバイスをフォーマットするには、次のコマンドを入力します。

```
[local]host_name# filesystem format { /flash | /usb1 | /hd-raid }
```

既存の CLI コンフィギュレーション ファイルの適用

既存の CLI 設定ファイルは、ユーティリティ機能（テスト時のすべての統計情報のクリアなど）を提供するために作成されたか、またはテキストエディアを使用してオフラインで作成された .conf ファイルです。既存の設定ファイルはローカルファイルシステムに保存されていることがあり、そのファイルはいつでも実行中のシステムに適用できます。

注意 現在別の CLI 設定を実行しているシステムに設定ファイルを適用したすると、同じコマンドが適用しようとしている設定ファイルに存在する場合は、類似するコンテキスト、論理インターフェイス、物理ポート、IP アドレス、またはその他の設定済みの項目はすべて上書きされます。適用しようとしているファイルの内容を十分に理解し、現在実行中のコマンドが上書きされた場合はサービスにどのような影響があるかを把握しておく必要があります。また、変更は自動的に保存されないことにも注意してください。

CLI 設定ファイル、または CLI コマンドを含むスクリプトは、Exec モードプロンプトで次のコマンドを入力することによって実行中のシステムに適用できます。

```
[local]host_name# configure url [ verbose ]
```

url は、適用する CLI 設定ファイルの場所を指定します。ローカルファイルまたはリモートファイルを参照する場合があります。

次のコマンドは、 */flash* ディレクトリ内の *clearcmds* という名前の既存の CLI 設定ファイルを適用します。

```
[local]host_name# configure /flash/clearcmds.cfg
```

ローカルファイルシステム上のファイルの表示

この項では、さまざまなファイルを表示する方法について説明します。

ローカルデバイスの内容の表示

任意のローカルデバイスのコンテンツ、使用状況情報、およびファイルシステムディレクトリ構造を表示するには、Exec モードプロンプトで次のコマンドを入力します。

directory { /flash | /usb1 | /hd-raid }

CLI 設定と boot.sys ファイルの表示

ローカルファイルシステムに格納されている CLI 設定ファイルと boot.sys ファイルの内容は、Exec モードのプロンプトで次のコマンドを入力することによって、オフラインで表示できます（OS にロードする必要はありません）。

[local]*host_name*# **show file** *url* **{ /flash | /usb1 | /hd-raid }** *filename*

ここで、*url* はファイルの場所のパス名、*filename* は拡張子を含むファイルの名前を指します。

重要 オペレータレベルのユーザーとインスペクタレベルのユーザーは **show file** コマンドを実行できますが、 **directory** コマンドを実行することはできません。

オペレーティングシステムのファイルの検証

.bin 拡張子で識別されるオペレーティングシステムのソフトウェアイメージファイルは、システム上で実行され、ランタイムオペレーティングシステム（OS）を作成する、読み取り不可能かつ編集不可能なファイルです。

新しいオペレーティング システム イメージ ファイルをロードする前に確認することが重要です。これを実現するために、独自のチェックサムアルゴリズムを使用して、プログラムのコンパイル中に.bin ファイル内に保存されているアプリケーションの各部分のチェックサム値を作成します。

この情報は、コンパイル中にファイル内に保存されているチェックサム値に対して実際のファイルを検証するために使用できます。イメージファイルのいずれかの部分が破損した場合（たとえば、ファイルが切り捨てられた場合や、バイナリモードではなく ASCII モードを使用して転送された場合など）、この情報が報告され、ファイルは使用できないと見なされます。

オペレーティングシステムのソフトウェアイメージファイルを検証するには、Exec モードプロンプトで次のコマンドを入力します。

[local]*host_name*# **show version { /flash | /usb1 | /hd-raid }**
/*[directory]***/***filename* **[all]**

このコマンドの出力には、次の情報の列が表示されます。

- バージョン番号

- 説明
- 日付（Date）
- 起動イメージ
- サイズ
- フラグ
- プラットフォーム

無効なファイルが見つかった場合、システムは次のようなエラーメッセージを表示します。

```
Failure: Image /flash/image_version.bin CRC check failed!
Failure: /flash/image_version.bin, has a bad magic number
```

Elastic Services Controller のクラウド初期化サポート

Elastic Services Controller（ESC）が QvPC-DI の制御機能（CF）にアクティブおよびスタンバイで Cinder マルチアタッチボリュームを使用する場合、仮想マシン（VM）オーケストレーション用に呼び出される Openstack API バージョンは 2.60 以上です。この API バージョンでは、ESC はセキュリティ上の理由から、構成ファイルをエンコードして VM に挿入します。VM はエンコードされた構成ファイルを読み取ることができないため、ESC は **user_data** 圧縮ファイルを使用します。この **user_data** ファイルには、VM の起動に必要な構成ファイルが含まれています。

起動スタックの設定

起動スタックは、オペレーティングシステムのソフトウェアイメージと CLI 設定ファイルとの関連付けに優先順位を付けたリストで構成されます。これらの関連付けによって、システムの起動時またはリロード/再起動時にロードされるソフトウェアイメージと設定ファイルが決まります。関連付けは複数設定できますが、システムが使用するのは最も高い優先順位を持つ関連付けです。この関連付けの処理中にエラーが発生した場合（いずれかのファイルが見つからないなど）、システムは次に高い優先順位の関連付けを使用しようとします。

VPC-SI プラットフォームと VPC-DI プラットフォームでは、設定されたブート優先順位が最も高い構成ファイルが利用できない場合（ただし、イメージファイルは利用可能）、システムは次に利用可能なブートシステムの優先順位を使用する代わりに、リロード後に設定ウィザードを使用して起動します。優先順位は 1 ～ 100 で、1 が最も高い優先順位です。boot.sys ファイル内に設定できる起動スタックエントリの最大数は 10 です。

起動スタック情報は、boot.sys ファイルの概要 （2 ページ）で説明されているように、boot.sys ファイルに含まれています。次の項で説明するように、boot.sys ファイルには、起動スタックエントリの他に、システムの起動方式を定義するために必要なすべての設定コマンドが含まれています。

システムの起動方式

ローカル起動方式では、システムにローカルに保存されているソフトウェアイメージと設定ファイルを使用します。システムの起動時または再起動時に、ローカルデバイスのいずれか、または**/hd-raid**で特定のソフトウェアイメージとそれに付随する設定テキストファイルを調べます。ローカル起動方式を使用している場合は、起動スタックパラメータの設定のみが必要です。

システムは、特定の外部ネットワークサーバーからシステムに存在する設定テキストファイルとペアになっているソフトウェアイメージを取得します。ネットワーク起動を使用する場合は、次を設定する必要があります。

- 起動スタックパラメータ。これらは使用するファイルとそれらに使用する優先順位を定義します。
- リモート管理 LAN インターフェイスを定義する起動インターフェイスおよびネットワークパラメータと、外部ネットワークサーバーに到達するために使用する方式
- 確立するネットワーク通信を可能にするための遅延期間（秒単位）を定義するネットワーク起動遅延時間およびオプションのネームサーバーパラメータと、使用される可能性があるドメインネームサービス（DNS）ネームサーバーの IP アドレス

現在の起動スタックの表示

boot.sys ファイルに含まれている起動スタックのエントリを表示するには、Exec モードの**show boot** コマンドを実行します。

重要 **show boot** コマンドはオペレータレベルのユーザーとインスペクタレベルのユーザーが実行できます。

次に、ローカル起動設定のコマンド出力の例を示します。これらの例では、イメージファイル（オペレーティングシステム ソフトウェア）と設定ファイル（CLI コマンド）の両方が、**/flash** デバイスに配置されていることに注意してください。

重要 StarOS イメージファイル名の形式は「asr5500-*image_number*.bin」です。

例：

```
boot system priority 18 \
    image /flash/16-1-builds/asr5500-16.1.3.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/16-1-builds/asr5500-16.1.1.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
```

```
    image /flash/16-1-builds/asr5500-16.1.0.bin \
    config /flash/general_config_3665.cfg
```

次に、ネットワーク起動とローカル起動を組み合わせた設定の出力の例を示します。この例では、イメージファイル（オペレーティング システム ソフトウェア）は最初の2つの起動スタックエントリ（優先順位 18 と 19）により Trivial File Transfer Protocol（TFTP）を使用して外部ネットワークサーバーからロードされますが、すべての設定ファイルが **/flash** に配置されます。

また、起動スタックの上部にある起動ネットワーク インターフェイスと起動ネットワーク設定のコマンドにも注意してください。これらのコマンドは、使用するリモート管理LANインターフェイスと、オペレーティング システム ソフトウェアのイメージファイルをホストする外部ネットワークサーバーとの通信に関する方法を定義します。

```
boot networkconfig static ip address mio1 192.168.1.150 netmask 255.255.255.0
boot delay 15
boot system priority 18 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 19 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 20 image /flash/image_version.bin \config /flash/general_config.cfg
```

初期起動時にロードされた起動イメージの優先順位を確認するには、次のように入力します。

show boot initial-config

次に、出力例を示します。

```
[local]host_name# show boot initial-config
Initial (boot time) configuration:
      image tftp://192.168.1.161/tftpboot/image_version.bin \
       config /flash/config_name.cfg
      priority 1
```

新しい起動スタックエントリの追加

重要 この手順を実行する前に、boot.sys ファイルに10未満のエントリがあり、より高い優先順位のエントリが使用可能であることを確認します（つまり、少なくとも起動スタックに優先順位 1 のエントリがないことを確認します）。詳細については、「現在の起動スタックの表示」を参照してください。

優先順位1が使用されている場合は、既存のエントリの番号を付け直して、少なくともその優先順位が使用可能であることを確認する必要があります。*boot.sys* ファイルに含めることができる起動スタックエントリの最大数は10です。起動スタックにすでに10個のエントリがある場合は、これらのエントリのうち少なくとも1つを削除する必要があります（通常は優先順位が最も低いエントリ）。また、必要に応じて、他のエントリの一部またはすべての番号を再割り当てしてから続行します。詳細については、起動スタックエントリの削除 （10 ページ） を参照してください。

この手順では、新しい起動スタックエントリを *boot.sys* ファイルに追加する方法について説明します。Exec モードのプロンプトが表示されていることを確認し、次のコマンドを入力します。

```
configure
 boot system priority number image image_url config cfg_url
```

次のコマンドは、起動優先順位 *3* を使用して、新しい起動スタックエントリを作成します。

```
boot system priority 3 image /flash/image_filename.bin config
/flash/config_name.cfg
```

重要 boot.sys ファイルに保存された起動スタックの変更は、システムが再起動されるまで実行されません。

次のコマンドを使用して、CF VM 上のローカルファイルシステムを同期します。

```
filesystem synchronize all
```

起動スタックエントリの削除

この手順では、boot.sys ファイルから個々の起動スタックエントリを削除する方法について説明します。Exec モードのプロンプトが表示されていることを確認し、次のコマンドを入力します。

```
configure
  no boot system priority number
```

number は、起動スタックエントリに使用される起動の優先順位を指定します。このコマンドは、起動スタックから特定のエントリを削除して、 boot.sys ファイルが上書きされるようにします。

オペレーティングシステムソフトウェアのアップグレード

この項では、StarOS バイナリイメージのアップグレード手順を手動で実行する方法について説明します。

StarOS ソフトウェアのアップグレード手順を開始する前に、「前提条件」の項で説明されている条件が満たされていることを確認してください。

注意 VPC の展開解除/再展開は、bin アップグレード後はサポートされません。VPC を非アクティブにすると、アップグレードした StarOS の bin イメージが削除されます。

StarOS ソフトウェアを手動でアップグレードするには、次の手順を実行します。

1. *AutoVNF*、*CF*、*ESC*、および *UEM* の *VIP* アドレスの取得
2. OS リリースバージョンとビルド番号の識別 （13 ページ）
3. サポートサイトからソフトウェアイメージをダウンロード （14 ページ）
4. *Zookeeper* データベースの確認
5. *ESC* データベースの確認
6. /flash デバイスの空き領域の確認 （16 ページ）
7. StarOS イメージを /flash に転送 （17 ページ）
8. 実行コンフィギュレーションの保存 （22 ページ）
9. ファイルシステムの同期 （30 ページ）
10. システムの再起動 （24 ページ）

前提条件

CF と SF の VNFC を含む StarOS ソフトウェアのアップグレードを実行する前に、次の前提条件が満たされているかどうかを確認します。

- AutoDeploy、AutoVNF、ESC、UEM、および CF VM のログインクレデンシャルと IP アドレスが必要です。OpenStack 設定の管理権限を持っている必要があります。
- OpenStack のステータスを確認します。Ansible の出力がすべて引き渡される必要があります。

```
cd /home/stack/
source stackrc
cd /home/stack/ansible/
ansible-playbook -i inventory openstack_verify.yml
```

- AutoVNF/ESC/EM/VNF VM の正常性が AutoIT の UltraM 正常性ログを通じて正常であるかどうかを確認します。いずれかの VM が正常でない場合は、対応する VM の正常性を回復するための必要なアクションを実行します。
- 新しい StarOS バイナリイメージファイル（手動アップグレードの場合） が必要です。
- ESC、UEM、および CF の間に保留中のトランザクションがないことを確認します。
- 元の StarOS bin ファイルのバックアップを必ず実行します。

AutoVNF、CF、ESC、および UEM の VIP アドレスの取得

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

AutoVNF、CF、ESC、および UEM VM の VIP アドレスの収集

1. デフォルトのユーザーである *ubuntu* として AutoDeploy VM にログオンします。
   ssh ubuntu@*<ad_vm_address>*
2. ルートユーザーに切り替えます。
   sudo -i
3. ConfD CLI を入力します。
   confd_cli -u admin -C
4. プロンプトが表示されたら、管理者ユーザーのログイン情報を入力します。
5. AutoVNF、ESC、UEM、および CF VM の VIP アドレスを収集します。
   show vnfr
   出力例：

```
vnfr autoit-f-autovnf
 vnfd     f-autovnf
 vnf-type usp-uas
 state    deployed
 external-connection-point avf
  virtual-link-ref    management
  ip-address          192.168.100.26
  floating-ip-address 10.225.202.94
vnfr sj-autovnf-esc
 vnfd     esc
 vnf-type esc
 state    deployed
 external-connection-point esc
  virtual-link-ref management
  ip-address       192.168.100.22
vnfr sj-autovnf-vpc
 vnfd     vpc
 vnf-type ugp
 state    alive
 external-connection-point cf
  virtual-link-ref management
```

```
 ip-address       192.168.100.38
external-connection-point em
 virtual-link-ref management
 ip-address       192.168.100.21
```

OS リリースバージョンとビルド番号の識別

オペレーティングシステムは、CLI から発行されたコマンドを使用して、サービスを提供し、事前定義された機能を実行するように設定できます。

オペレーティングシステムのソフトウェアは、単一のバイナリファイル（ファイル拡張子 **.bin**）として提供され、システム全体の単一インスタンスとしてロードされます。

- イメージファイル名は、プラットフォームタイプとリリース番号を指定するサフィックスで識別されます。たとえば、**asr5500-***release_number***. bin** というようになります。たとえば、 **asr5500-16.1.0.bin** というようになります。

starfile イメージは、リリース前に REL キーで署名する必要があります。展開可能なイメージは、「.bin.SPA」拡張子を持つ REL キーを使用して署名されます。「A」は、署名キーのリビジョンレベルを示します。たとえば、**asr5500-20.0.0.bin.SPA** というようになります。署名キーが侵害されると、新しいキーが作成され、リビジョンレベルが「B」に増加します。

信頼できるイメージが導入されました。信頼できるビルドと通常のビルドの違いは、非セキュアなプログラムである ftpd、telnet および tcpdump がないことと、セキュリティオプション用の staros.conf ファイルが追加されていることです。信頼できるイメージは、プラットフォーム名に「_T」が存在することによって識別できます。たとえば、**asr5500_T-20.0.0.bin.SPA** というようになります。

StarOS ソフトウェアのバージョンとビルド情報を確認するには、次のようにします。

1. アップグレードする VNF にログオンします。
2. StarOS コマンド ライン インターフェイスで次の Exec モードのコマンドを入力します。

   show version

   出力例：

```
Active Software:
  Image Version:             21.9.0.69918
  Image Build Number:        69918
  Image Description:         Deployment_Build
  Image Date:                Sun Jul 22 12:08:55 EDT 2018
  Boot Image:                /flash/staros.bin
  Source Commit ID:          94797337b6c1691541ea0dd86f2f29b0f2c3630c
```

3. StarOS ビルドリリースに関する追加情報を表示するには、次の Exec モードのコマンドを実行します。

   show build

サポートサイトからソフトウェアイメージをダウンロード

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

シスコのサポートサイトとダウンロードファシリティへのアクセスは、ユーザー名とパスワードで制御されています。サイトにアクセスして StarOS のイメージをダウンロードするには、アクティブなカスタマーアカウントが必要です。

/flash デバイスにアップロードできるネットワークの場所または物理デバイス（USB スティック）に、ソフトウェアイメージをダウンロードします。詳細についてはシスコの担当者または Cisco TAC にお問い合わせください。

UGP ベースの VNF の場合、次の手順を実行して、新しい bin ファイルを AutoVNF または OSPD VM にダウンロードします。

1. 対応する VNF の AutoVNF にログオンします。

   ssh ubuntu@*<ad_vm_address>*

   コマンドの例：

   ssh ubuntu@10.225.202.94

2. 新しい StarOS qvpc-di バイナリファイルを AutoVNF/OSPD にダウンロードするためのディレクトリを作成します。

   cd /home/ubuntu/

   mkdir StarOSBinUpgrade

3. Cisco のサポートサイトから新しい StarOS qvpc-di バイナリファイルをダウンロードし、*StarOSBinUpgrade* ディレクトリにファイルをコピーします。

   cd StarOSBinUpgrade

   次のコマンドを使用して、ディレクトリに新しい bin ファイルが含まれているかどうかを確認します。

   ls -lrt /home/ubuntu/StarOSBinUpgrade

   出力例：

   ```
   total 172560

   -r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
   ```

Zookeeper データベースの確認

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

zookeeper データベースを確認するには、次のようにします。

1. フローティング IP を使用して AutoVNF にログオンします。

   ssh ubuntu@*<ad_vm_address>*

   コマンドの例：

   ssh ubuntu@10.225.202.94

2. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で取得した VIP アドレスを使用して、UEM VM にログオンします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

   ssh ubuntu@192.168.100.21

3. ルートユーザーになります。

   sudo -i

4. Zookeeper データベース接続の UEM オーケストレーション IP アドレスを収集します。

   #ifconfig

   ```
   eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

             inet addr:209.165.200.240  Bcast: 209.165.200.255  Mask: 255.255.255.224
   ```

5. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* ディレクトリに移動します。

6. UEM Zookeeper データベースにアクセスするには、コマンドラインから次のスクリプトを実行します。

   zkCli.sh -server *ip_addr:port_num*

   次に例を示します。

   zkCli.sh -server 209.165.200.240:2181

7. zookeeper データベースを確認し、UEM VM と CF VM の間に保留中の要求がないことを確認します。

   ls /request

   出力例：

   ```
   []

   <Ctrl+D to exit Zookeeper shell>
   ```

ESC データベースの確認

この項では、CF および SF VNFC のアップグレードにのみ適用される手順について説明します。

ESC データベースを確認するには、次のようにします。

1. フローティング IP を使用して AutoVNF にログオンします。

   ssh ubuntu@*<ad_vm_address>*

   コマンドの例：

   ssh ubuntu@10.225.202.94

2. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で取得した VIP アドレスを使用して ESC VM にログオンします。

   ssh admin@*<vip-addr>*

   コマンドの例：

   ssh admin@192.168.100.22

3. ESC データベースを確認して、保留中のトランザクションがないことを確認します。

   sudo /opt/cisco/esc/pgsql/bin/psql -U esc -p 7878 -h localhost -c 'select * from esc_schema.workitem';

   ```
   config_id | request_id | mo_type | config_action | config_state
   ----------+------------+---------+---------------+--------------
   (0 rows)
   ```

4. トランザクションの詳細を確認するには、次のコマンドを実行します。

   escadm ip_trans

   出力例：

   ```
   Number of in-progress transaction events = 0
   ```

/flash デバイスの空き領域の確認

新しい StarOS イメージファイルに対応するために、**/フラッシュ**デバイスに十分な空き領域があることを確認します。

フラッシュディレクトリで使用可能な領域を確認するには、次のようにします。

1. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で以前に取得した VIP アドレスを使用して CF VM にログインします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

   ssh ubuntu@192.168.100.38

2. 次の Exec モードコマンドを入力します。

```
[local]host_name# directory /flash
```

次に、表示されるディレクトリ情報のタイプの例を示します。

```
-rwxrwxr-x    1  root     root         7334 May    5 17:29 asr-config.cfg
-rwxrwxr-x    1  root     root         399 Jun    7 18:32 system.cfg
-rwxrwxr-x    1  root     root         10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x    1  root     root         10667 Jun    1 11:21 testconfig_4.cfg
-rwxrwxr-x    1  root     root         5926 Apr    7 16:27  tworpcontext.cfg
-rwxrwxr-x    1  root     root         15534 Aug    4 13:31 test_vlan.cfg
-rwxrwxr-x    1  root     root         2482 Nov 18 11:09 gateway2.cfg
-rwxrwxr-x    1  root     root         159106048 Dec 31  2011 image_filename
1136352  /flash
Filesystem           1k-blocks      Used Available Use% Mounted on
/var/run/storage/flash/part1 3115468   1136352  30018336   4%
/mnt/user/.auto/onboard/flash
```

ディスプレイの最後の行にある「Available」ブロックに注意してください。ディレクトリ情報を表示した後、CLI はルートに戻り、次のプロンプトが表示されます。

```
[local]host_name#
```

StarOS イメージを /flash に転送

次のいずれかの方法を使用して、新しいオペレーティングシステムのイメージファイルを MIO/UMIO/MIO2 VPC-DI アクティブ CF または VPC-SI 上の */flash* ディレクトリに転送します。

- システムへのアクセス権を持つ FTP クライアントを使用して、ファイルを */flash* デバイスに転送します。

重要 ファイル転送プロトコル（FTP）を使用してオペレーティングシステムのソフトウェアイメージファイルを転送する場合は、バイナリモードを使用してファイルを転送するように FTP クライアントを設定する必要があります。バイナリ転送モードを使用しないと、転送されたオペレーティング システム イメージファイルが使用できなくなります。FTP はサポートされていません。

- システムへのアクセス権を持つ SFTP クライアントを使用して、ファイルを */flash* デバイスに転送します。

UGP ベースの VNF の場合は、次の手順に従って、新しい StarOS bin をアクティブ CF にコピーします。

1. 新しい bin ファイルがダウンロードされた AutoVNF または OSPD VM にログオンします。

   ssh ubuntu@*<ad_vm_address>*

   コマンドの例：

   ssh ubuntu@10.225.202.94

2. 新しい bin ファイルがシスコのサポートサイトからダウンロードされたディレクトリに移動します。

   cd /home/ubuntu/StarOSBinUpgrade/ && ls -lrt

   出力例：

   ```
   total 172560

   -r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
   ```

3. SFTP から CF VM へ。

   次に例を示します。

   sftp ubuntu@192.168.100.38

4. *sftp* ディレクトリに移動します。

   ```
   #sftp>pwd

   Remote working directory: /

   #sftp>ls

   hd-raid sftp

   #sftp>cd sftp
   ```

5. 新しいバイナリファイルを *sftp* ディレクトリにアップロードします。

   ```
   #sftp>put image_filename.bin
   ```

   出力例：

   ```
   #sftp>put qvpc-di-21.9.0.69932.bin

   Uploading qvpc-di-21.9.0.69932.bin to
   /.auto/onboard/flash/sftp/qvpc-di-21.9.0.69932.bin
   qvpc-di-21.9.0.69932.bin                          100%  169MB 168.5MB/s   00:01
   ```

6. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で取得した VIP アドレスを使用して CF VM にログオンします。

   ssh ubuntu@<*vip-addr*>

   コマンドの例：

   ssh ubuntu@192.168.100.38

7. 新しい bin を *sftp* から *flash* ディレクトリにコピーします。

   copy /flash/sftp/*image_filename.bin* **/flash/updated.bin**

   出力例：

   ```
   #copy /flash/sftp/qvpc-di-21.9.0.69932.bin /flash/updated.bin

   ************************************************************

   Transferred 176698880 bytes in 2.718 seconds (63486.9 KB/sec)
   ```

8. *sftp* ディレクトリから新しい bin を削除します。

   delete /flash/sftp/*image_filename.bin*

出力例：

```
delete /flash/sftp/qvpc-di-21.9.0.69932.bin
Are you sure? [Yes|No]: yes
File /flash/sftp/qvpc-di-21.9.0.69932.bin removed
```

9. 次の Exec モードのコマンドを実行して、イメージファイルが */flash* デバイスに正常に転送されたことを確認します。

   [local]*host_name*# **directory /flash**

   表示された出力にイメージファイル名が表示されます。

10. 次のコマンドを実行して、ビルド情報を確認します。

    show version /flash/*image_filename***.bin**

現在の設定ファイルのコピーの保存

新しいソフトウェアリリースにアップグレードする前に、現在の設定ファイルを **/flash** デバイスとシャーシ外の場所（外部メモリデバイスやネットワーク URL）にコピーして名前を変更する必要があります。この名前が変更されたコピーが、アップグレード中に問題が発生した場合に使用できるフォールバック用のロードが可能な設定ファイルとなります。

リリース 20.0 からのダウングレード

PBKDF2（パスワードベースのキー派生関数バージョン 2）を使用し、入力されたデータ、ソルト、および反復回数に基づいて、特定の長さのキーを取得するようになりました。ローカルユーザーアカウントのパスワードは、ランダムに生成されたソルトと多くの反復回数を備えた PBKDF2 方式を使用してハッシュされ、パスワードストレージの保護が強化されます。

MD5 ハッシュアルゴリズムを使用するようにローカルユーザーデータベースをダウングレードするには、セキュリティ管理者が Exec モードの **downgrade local-user database** コマンドを実行する必要があります。StarOS は確認のプロンプトを表示し、パスワードを再入力するようセキュリティ管理者に要求します。ダウングレードコマンドを実行する前に、ユーザーは入力したパスワードで再認証されます。確認後、パスワードは適切な古いまたは弱い暗号化アルゴリズムを使用してハッシュされ、データベースに保存されて、以前のバージョンの StarOS がセキュリティ管理者を認証できるようになります。

ダウングレードプロセスでは、PBKDF2 でハッシュされたパスワードは MD5 形式に変換されません。ダウングレードプロセスでは、（/flash ディレクトリから）データベースが再度読み込まれ、古い形式でデータベースが再構築されてからディスクに書き込まれます。PBKDF2 でハッシュされたパスワードは MD5 ハッシュアルゴリズムに変換できず、以前の StarOS リリースでは PBKDF2 暗号化アルゴリズムを解析できないため、StarOS は PBKDF2 アルゴリズムを介して暗号化されたすべてのユーザーを一時停止します。MD5 アルゴリズム（「弱いハッシュ」フラグ）を介して暗号化されたユーザーは、ログイン情報を使用してログインを続行できます。システムが以前の StarOS リリースで起動した後、一時停止されたユーザーは **show local-user [verbose]** コマンドの出力で確認できます。

一時停止されたユーザーを再アクティブ化するには、セキュリティ管理者が次の手順を実行します。

- Exec モードの **password change local-user** *username* コマンドを使用して、一時停止されたユーザーの一時パスワードを設定します。
- コンフィギュレーション モードの **no suspend local-user** *username* コマンドを使用して、ユーザーの一時停止フラグをリセットします。

オフラインソフトウェアのアップグレード

オフラインソフトウェアのアップグレードは、バージョン番号に関係なく、任意のバージョンのオペレーティングシステムのソフトウェアから、任意のバージョンにアップグレードするすべてのシステムに対して実行できます。このプロセスは、システムが現在のセッションをサポートしている間に多くのステップを実行できますが、このプロセスの最後のステップでは、実際にソフトウェアのアップグレードを適用するために再起動する必要があるので、オフラインと見なされます。

この手順では、CLIセッションが確立されており、新しいオペレーティングシステムのイメージファイルをローカルファイルシステムに配置していることを前提としています。開始するには、Exec モードのプロンプトが表示されていることを確認してください。

```
[local]host_name#
```

オフラインソフトウェアのアップグレードを実行するには：

1. 新規コールポリシーの設定 （20 ページ）
2. Day バナーのメッセージの設定 （21 ページ）
3. 現在の CLI コンフィギュレーション ファイルのバックアップ （21 ページ）
4. 実行コンフィギュレーションの保存 （22 ページ）
5. 新しい起動スタックエントリの作成 （23 ページ）
6. ファイルシステムの同期 （30 ページ）
7. システムの再起動 （24 ページ）

新規コールポリシーの設定

サービス要件を満たすために、Exec モードから新規コールポリシーを設定します。このポリシーを有効にすると、アップグレードプロセスが完了したシステムのリロードを予測して、新規コールがリダイレクトまたは拒否されます。これにより、アップグレードが完了したシステムのリロードによって発生するサブスクライバへのサービスの中断時間が短縮されます。

重要 新規コールポリシーは、サービス単位で作成されます。シャーシで複数のサービスを実行している場合は、複数の新規コールポリシーを設定できます。

新規コールポリシーのシンタックスについては、以下を参照してください。

```
[local]host_name# newcall policy { asngw-service | asnpc-service |
sgsn-service } { all | name service_name } reject
[local]host_name# newcall policy { fa-service | lns-service |
mipv6ha-service } { all | name service_name } reject
[local]host_name# newcall policy { ha-service | pdsn-service |
pdsnclosedrp-service  } { all | name service_name } { redirect
target_ip_address [ weight  weight_num ] [ target_ipaddress2 [ weight weight_num ]
 ... target_ip_address16 [ weight weight_num ] | reject }
[local]host_name# newcall policy ggsn-service { apn name apn_name | all |
name service_name } reject
[local]host_name# newcall policy hnbgw-service { all | name service_name }
reject
[local]host_name# newcall policy { pcc-af-service | pcc-policy-service } {
 all | name service_name } reject
[local]host_name# newcall policy {pcc-af-service | pcc-policy-service } {
all | name service_name } reject
[local]host_name# newcall policy mme-service { all | name service_name }
reject
```

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

Day バナーのメッセージの設定

オプション：グローバル コンフィギュレーション モードのプロンプトから次のコマンドを入力して、システムを再起動されることを他の管理ユーザーに通知する「Message of the Day」バナーを設定します。

```
[local]host_name(config)# banner motd "banner_text"
```

banner_text は、表示するメッセージで、最大 2048 文字の英数字を使用できます。*Banner_text* は、引用符で始める引用符で終わる（""）必要があることに注意してください。CLI バナー情報の入力の詳細については、『*CLI Reference*』を参照してください。バナーは、管理ユーザーが CLI にログインしたときに表示されます。

現在の CLI コンフィギュレーション ファイルのバックアップ

次のコマンドを入力して、現在の CLI コンフィギュレーション ファイルをバックアップします。

```
[local]host_name# copy from_url to_url [ -noconfirm ]
```

これにより、現在の起動スタックエントリで定義されているオペレーティングシステムにリンクされている CLI コンフィギュレーション ファイルのミラーイメージが作成されます。

次のコマンド例では、*/flash* デバイスにある **general.cfg というファイル**のバックアップコピーを、*general_3652.cfg* というファイルに作成します。

```
[local]host_name# copy /flash/general.cfg /flash/general_3652.cfg
```

実行コンフィギュレーションの保存

シャーシを再起動する前に、現在実行中のアップグレード済みの設定を保存します。

起動設定を保存するには、次を実行します。

1. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で以前に取得した VIP アドレスを使用して VNF にログオンします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

   ssh ubuntu@192.168.100.38

2. オプションです。Exec モードで次のコマンドを実行します。

   chassis key value 1234

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

   重要 この手順は任意であり、シャーシキーが設定されていない場合にのみ必要になります。

3. フラッシュディレクトリに起動設定を保存します。

   save configuration /flash/system.cfg

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   これにより、新しい bin イメージを使用するように起動設定が更新されます。

   次のコマンドを使用して起動設定を確認します。

   # show boot

   ```
   Monday May 21 20:39:57 UTC 2018

   boot system priority 8 \
       image /flash/sftp/production.YYYYY.qvpc-di.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 9 \
       image /flash/staros.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 10 \
       image /flash/staros.bin \
       config /flash/system.cfg
   ```

4. コンフィギュレーション モードを開始し、新しい StarOS bin ファイルの起動優先順位を変更します。

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. 新しい起動優先順位を確認します。

```
#show boot

boot system priority 1 \

    image /flash/updated.bin \

    config /flash/system.cfg

boot system priority 10 \

    image /flash/staros.bin \

    config /flash/system.cfg
```

6. フラッシュディレクトリに起動設定と新しい bin が含まれているかどうかを確認します。

```
dir /flash

total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

新しい起動スタックエントリの作成

次のグローバル コンフィギュレーション コマンドを入力して、新しいオペレーティングシステムイのメージファイルと現在使用されているCLI設定ファイルで構成される新しいファイルグループの新しい起動スタックエントリを作成します。

`[local]`*host_name*`(config)#` **boot system priority** *number* **image** *image_url* **/flash** *filename* **config** *cfg_url* **/flash/***filename*

<N-1>方式を使用して、このエントリに次に高い優先順位を割り当てます。この場合、優先順位番号は、現在の最高の優先順位よりも 1 つ小さい値を割り当てます。

重要 Exec モードの **show boot** コマンドを実行して、boot.sys ファイルに 10 未満のエントリがあることと、高い優先順位のエントリが使用可能である（最小で起動スタックに優先順位 1 のエントリがない）ことを確認します。

優先順位 1 が使用されている場合は、既存のエントリの数字を再割り当てし、少なくともその優先順位 1 を使用できるようにします。

boot.sys ファイルに含めることができる起動スタックエントリの最大数は 10 です。起動スタックにすでに 10 個のエントリがある場合は、これらのエントリのうち少なくとも 1 つを削除する必要があります（通常は優先順位が最も低いエントリ）。また、必要に応じて、他のエントリの一部またはすべての番号を再割り当てしてから続行します。起動スタックエントリを削除するには、no boot system priority コマンドを使用します。

```
[local]host_name# configure
[local]host_name(config)# no boot system priority number
```

新しい起動スタックエントリを boot.sys.sys ファイルに追加するには、次のコマンドを入力します。

```
[local]host_name# configure
[local]host_name(config)# boot system priority number image image_url config
cfg_url
```

boot system priority コマンドの使用方法については、新しい起動スタックエントリの追加 （9 ページ）を参照してください。

システムの再起動

システム（VNF）を再起動するには、次のようにします。

1. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で以前に取得した VIP アドレスを使用して VNF にログオンします。

   ssh ubuntu@<*vip-addr*>

   コマンドの例：

   ssh ubuntu@192.168.100.38

2. 次の Exec モードコマンドを入力します。

   ```
   [local]host_name# reload [-noconfirm]
   ```

   システムが再起動すると、以前に設定した新しい起動スタックエントリを使用して、新しいオペレーティングシステムのソフトウェアイメージとそれに対応する CLI コンフィギュレーション ファイルがロードされます。

3. *PDSN* のオプション：アップグレード中に IP プール共有プロトコルを使用している場合は、『*PDSN Administration Guide*』の「*Configuring IPSP Before the Software Upgrade*」を参照してください。

4. リロードが完了したら、VNF にログオンして、目的の StarOS バージョンでロードされていること、およびすべてのカードが起動していること、および予想どおりにアクティブ状態またはスタンバイ状態になっていることを確認します。

   show version

   出力例：

   ```
   Active Software:
     Image Version:                  21.9.0.69977
     Image Build Number:             69977
     Image Description:              Build
     Image Date:                     Mon Jul 30 06:48:34 EDT 2018
   ```

```
  Boot Image:                     /flash/updated.bin
  Source Commit ID:               abde005a31c93734c89444b8aec2b6bb2d2e794d
```

show card table

出力例：

```
Slot         Card Type                                Oper State     SPOF  Attach
-----------  ---------------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card            Active         No
 2: CFC      Control Function Virtual Card            Standby        -
 3: FC       4-Port Service Function Virtual Card     Standby        -
 4: FC       4-Port Service Function Virtual Card     Standby        -
 5: FC       4-Port Service Function Virtual Card     Standby        -
 6: FC       4-Port Service Function Virtual Card     Standby        -
 7: FC       4-Port Service Function Virtual Card     Standby        -
 8: FC       4-Port Service Function Virtual Card     Standby        -
 9: FC       4-Port Service Function Virtual Card     Standby        -
10: FC       4-Port Service Function Virtual Card     Standby        -
```

5. 次の Exec モードコマンドを実行して、実行中の StarOS ビルドリリースに関する追加情報を表示します。

   show build

6. オプションです。CF および SF VNFC の動作状態を確認します。

（注） このステップは、CF および SF VNFC をアップグレードする場合にのみ該当します。

1. 「*Zookeeper* データベースの確認」と「*ESC* データベースの確認」の項の手順を繰り返します。
2. フローティング IP を使用して UEM にログオンするか、または UEM VIP を使用して AutoVNF から UEM にログオンします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

   ssh ubuntu@192.168.100.21

3. ルートユーザーになります。

   sudo -i

4. Zookeeper データベース接続の UEM オーケストレーション IP アドレスを収集します。

   #ifconfig

   ```
   eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

             inet addr:209.165.200.225  Bcast:209.165.200.255  Mask:255.255.255.224
   ```

5. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* ディレクトリに移動します。
6. UEM Zookeeper データベースにアクセスするには、Zookeeper ツールを実行します。

zkCli.sh -server *<vip-addr>:port_num*

コマンドの例：

zkCli.sh -server 209.165.200.225:2181

UEM と CF の間に未処理の要求がないことを確認してください。

7. 次のコマンドを使用して、各 CF および SF の「状態」：「alive」を確認します。

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf2**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf2**

コマンドの例：

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2

8. コンソール出力で Alive 状態を確認します。

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1da1e1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"},{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b111e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0
```

（注） **CTRL+D** を使用して、zookeeper CLI を終了することができます。

9. UEM VM からルートユーザーとして、*ncs_cli* にログオンし、デバイスのライブステータスを確認します。

~$ sudo -i

ncs_cli -C -u admin

 # show devices device *device_name* **live-status**

コマンド出力に、各カードの正しい「状態」と「カード状態」が反映されていることを確認します。

出力例：

 # show devices device tb1-autovnf1_vpc-vpc-core-cf-nc live-status

```
<snip>
```

VNFC の参照	現在の状態	VNFC インスタンス ID	VDU の参照	カードタイプ	カードスロット ID	コア数	CPU 使用率	ディスク容量	開始時刻	稼働時間	NOVA 起動 CMD	ID	日時	状態から	状態へ
cf1	-	cf1	cf	[illegible]	1	-	-	-	-	-	-				
cf2	-	cf2	cf	[illegible]	2	-	-	-	-	-	-				
sf1	-	sf1	sf	[illegible]	3	-	-	-	-	-	-				
sf2	-	sf2	sf	[illegible]	4	-	-	-	-	-	-				

```
live-status vnfd sj-autovnf-vpc-abc

 version            6.0

 vnfm vim-tenant-name abc

 vnfm tenant-name abc

 vnfm ipaddr     192.168.100.22

 vnfm port       830

 vnfm username   ubuntu

 vnfm password   "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRe1f84byNakWEa9sJ8sY/
cwfFME3aG0UaBC\nvvNNAMkuXQI9Ksfu5IiQQ9ViWbbHwl6IEFQ="

 virtual-link vl-di-internal1

  auto-vnf-connection-ref di-internal1

 virtual-link vl-management
```

```
  auto-vnf-connection-ref management

 virtual-link vl-orchestration

  auto-vnf-connection-ref orchestration

 virtual-link vl-abc-vpc-svc

  auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

 vdu cf

  ssh-keygen         false

  vm-image           076c887a-a12c-4a0b-b4d6-b2d213f64b9e

  lifecycle-event-initialization staros_config.txt

   source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt

  lifecycle-event-initialization staros_param.cfg

   source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg

  ned cisco-staros-nc

   user        "$4$+HLzsE1kLJOeuoFWyOsmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

   password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="

   netconf

   port-number 830

  card-type          control-function

  usp-auto-vnf-id    sj-autovnf-vpc-abc-cf

  vnfc cf-vnfc-ugp

<snip>
```

実行コンフィギュレーションの保存

シャーシを再起動する前に、現在実行中のアップグレード済みの設定を保存します。

起動設定を保存するには、次を実行します。

1. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で以前に取得した VIP アドレスを使用して VNF にログオンします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

ssh ubuntu@192.168.100.38

2. オプションです。Exec モードで次のコマンドを実行します。

   chassis key value 1234

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

   

   重要 この手順は任意であり、シャーシキーが設定されていない場合にのみ必要になります。

3. フラッシュディレクトリに起動設定を保存します。

   save configuration /flash/system.cfg

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   これにより、新しい bin イメージを使用するように起動設定が更新されます。

   次のコマンドを使用して起動設定を確認します。

   # show boot

   ```
   Monday May 21 20:39:57 UTC 2018

   boot system priority 8 \
       image /flash/sftp/production.YYYYY.qvpc-di.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 9 \
       image /flash/staros.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 10 \
       image /flash/staros.bin \
       config /flash/system.cfg
   ```

4. コンフィギュレーション モードを開始し、新しい StarOS bin ファイルの起動優先順位を変更します。

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. 新しい起動優先順位を確認します。

   #show boot

   ```
   boot system priority 1 \

       image /flash/updated.bin \

       config /flash/system.cfg

   boot system priority 10 \
   ```

```
image /flash/staros.bin \

config /flash/system.cfg
```

6. フラッシュディレクトリに起動設定と新しい bin が含まれているかどうかを確認します。

   dir /flash

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

ファイルシステムの同期

ファイルシステムを同期するには、次の手順を実行します。

1. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ）で取得した VIP アドレスを使用して VNF にログオンします。

2. 次のコマンドを入力して、管理カード上のローカルファイルシステムを同期します。

   [local]*host_name*# **filesystem synchronize all**

   出力例：

```
Updating /flash/system.cfg

******************************************************************************

Updating /flash/updated.bin

******************************************************************************

Updating /flash/sftp/yang/cisco-staros-bulkstats-config.yang

******************************************************************************

Updating /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang

******************************************************************************

Updating /flash/sftp/yang/cisco-staros-bulkstats.yang

******************************************************************************

Updating /flash/sftp/yang/cisco-staros-cli-config.yang

******************************************************************************
```

```
Updating /flash/sftp/yang/cisco-staros-confd-config.yang
**************************************************************************************
Updating /flash/sftp/yang/cisco-staros-config.yang
**************************************************************************************
Updating /flash/sftp/yang/cisco-staros-exec.yang
**************************************************************************************
Updating /flash/sftp/yang/cisco-staros-kpi.yang
**************************************************************************************
Updating /flash/sftp/yang/cisco-staros-notif.yang
**************************************************************************************
Updating /flash/boot.sys
**************************************************************************************
12 updated on card 2
           /flash/system.cfg
           /flash/updated.bin
           /flash/sftp/yang/cisco-staros-bulkstats-config.yang
           /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang
           /flash/sftp/yang/cisco-staros-bulkstats.yang
           /flash/sftp/yang/cisco-staros-cli-config.yang
           /flash/sftp/yang/cisco-staros-confd-config.yang
           /flash/sftp/yang/cisco-staros-config.yang
           /flash/sftp/yang/cisco-staros-exec.yang
           /flash/sftp/yang/cisco-staros-kpi.yang
           /flash/sftp/yang/cisco-staros-notif.yang
           /flash/boot.sys
```

システムの再起動

システム（VNF）を再起動するには、次のようにします。

1. AutoVNF、CF、ESC、および UEM の VIP アドレスの取得 （11 ページ） で以前に取得した VIP アドレスを使用して VNF にログオンします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

ssh ubuntu@192.168.100.38

2. 次の Exec モードコマンドを入力します。

   [local]*host_name*# **reload [-noconfirm]**

   システムが再起動すると、以前に設定した新しい起動スタックエントリを使用して、新しいオペレーティングシステムのソフトウェアイメージとそれに対応する CLI コンフィギュレーション ファイルがロードされます。

3. *PDSN* のオプション：アップグレード中に IP プール共有プロトコルを使用している場合は、『*PDSN Administration Guide*』の「*Configuring IPSP Before the Software Upgrade*」を参照してください。

4. リロードが完了したら、VNF にログオンして、目的の StarOS バージョンでロードされていること、およびすべてのカードが起動していること、および予想どおりにアクティブ状態またはスタンバイ状態になっていることを確認します。

   show version

   出力例：

```
Active Software:
  Image Version:                 21.9.0.69977
  Image Build Number:            69977
  Image Description:             Build
  Image Date:                    Mon Jul 30 06:48:34 EDT 2018
  Boot Image:                    /flash/updated.bin
  Source Commit ID:              abde005a31c93734c89444b8aec2b6bb2d2e794d
```

   show card table

   出力例：

```
Slot         Card Type                                Oper State     SPOF  Attach
-----------  ---------------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card            Active         No
 2: CFC      Control Function Virtual Card            Standby        -
 3: FC       4-Port Service Function Virtual Card     Standby        -
 4: FC       4-Port Service Function Virtual Card     Standby        -
 5: FC       4-Port Service Function Virtual Card     Standby        -
 6: FC       4-Port Service Function Virtual Card     Standby        -
 7: FC       4-Port Service Function Virtual Card     Standby        -
 8: FC       4-Port Service Function Virtual Card     Standby        -
 9: FC       4-Port Service Function Virtual Card     Standby        -
10: FC       4-Port Service Function Virtual Card     Standby        -
```

5. 次の Exec モードコマンドを実行して、実行中の StarOS ビルドリリースに関する追加情報を表示します。

   show build

6. オプションです。CF および SF VNFC の動作状態を確認します。

（注） このステップは、CF および SF VNFC をアップグレードする場合にのみ該当します。

1. 「*Zookeeper*データベースの確認」と「*ESC*データベースの確認」の項の手順を繰り返します。
2. フローティング IP を使用して UEM にログオンするか、または UEM VIP を使用して AutoVNF から UEM にログオンします。

   ssh ubuntu@*<vip-addr>*

   コマンドの例：

   ssh ubuntu@192.168.100.21
3. ルートユーザーになります。

   sudo -i
4. Zookeeperデータベース接続のUEMオーケストレーションIPアドレスを収集します。

   #ifconfig

   ```
   eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

             inet addr:209.165.200.225  Bcast:209.165.200.255  Mask:255.255.255.224
   ```
5. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* ディレクトリに移動します。
6. UEM Zookeeper データベースにアクセスするには、Zookeeper ツールを実行します。

   zkCli.sh -server *<vip-addr>:port_num*

   コマンドの例：

   zkCli.sh -server 209.165.200.225:2181

   UEM と CF の間に未処理の要求がないことを確認してください。
7. 次のコマンドを使用して、各 CF および SF の「状態」：「alive」を確認します。

   get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf1**

   get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf2**

   get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf1**

   get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf2**

   コマンドの例：

   get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1

   get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2

   get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1

   get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2
8. コンソール出力で Alive 状態を確認します。

   ```
   zk: localhost:2181(CONNECTED) 2] get
   /oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
   {"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
   ```

"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1da1e1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"},{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b111e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0

（注） **CTRL+D** を使用して、zookeeper CLI を終了することができます。

9. UEM VM からルートユーザーとして、*ncs_cli* にログオンし、デバイスのライブステータスを確認します。

   ~$ sudo -i

   ncs_cli -C -u admin

   # show devices device *device_name* **live-status**

   コマンド出力に、各カードの正しい「状態」と「カード状態」が反映されていることを確認します。

   出力例：

   # show devices device tb1-autovnf1_vpc-vpc-core-cf-nc live-status

   ```
   <snip>
   ```

VNFC の参照	現在の状態	VNFC インスタンス ID	VDU の参照	カードタイプ	カードスロット ID	コア数	CPU 使用率	ディスク容量	開始時刻	稼働時間	NOVA 起動 CMD	ID	日時	状態から	状態へ
cf1	-	cf1	cf	[illegible]	1	-	-	-	-	-	-				
cf2	-	cf2	cf	[illegible]	2	-	-	-	-	-	-				

```
 sf1  -    sf1  sf    active 3    -    -    -    -    -    -

 sf2  -    sf2  sf    active 4    -    -    -    -    -    -


live-status vnfd sj-autovnf-vpc-abc

 version            6.0

 vnfm vim-tenant-name abc

 vnfm tenant-name abc

 vnfm ipaddr     192.168.100.22

 vnfm port       830

 vnfm username   ubuntu

 vnfm password   "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRe1f84byNakWEa9sJ8sY/
cwfFME3aG0UaBC\nvvNNAMkuXQI9Ksfu5IiQQ9ViWbbHwl6IEFQ="

 virtual-link vl-di-internal1

  auto-vnf-connection-ref di-internal1

 virtual-link vl-management

  auto-vnf-connection-ref management

 virtual-link vl-orchestration

  auto-vnf-connection-ref orchestration

 virtual-link vl-abc-vpc-svc

  auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

 vdu cf

  ssh-keygen         false

  vm-image           076c887a-a12c-4a0b-b4d6-b2d213f64b9e

  lifecycle-event-initialization staros_config.txt

   source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt

  lifecycle-event-initialization staros_param.cfg

   source-url
http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg

  ned cisco-staros-nc

   user        "$4$+HLzsE1kLJOeuoFWyOsmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

   password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="
```

```
 netconf
 port-number 830
card-type           control-function
usp-auto-vnf-id     sj-autovnf-vpc-abc-cf
vnfc cf-vnfc-ugp
<snip>
```

以前のソフトウェアイメージの復元

何らかの理由でアップグレードを元に戻す必要がある場合は、次の場合を除き、アップグレードを再度実行します。

- アップグレード ソフトウェア イメージと設定ファイルの場所を指定します。

次に

- 元のソフトウェアイメージと設定ファイルの場所を指定します。

ライセンスキーの管理

ライセンスキーは、キャパシティ制限（許可されるサブスクライバセッションの数）とシステムで使用可能な機能を定義します。新しいライセンスキーを追加すると、キャパシティを増やしたり、サブスクライバベースが増加したときに新しい機能を追加したりすることができます。

新しいシステムライセンスキー

新しいシステムは、ライセンスキーがインストールされていない状態で納品されます。ほとんどの場合、ライセンスキーは電子形式（通常は電子メール）で渡されます。

ライセンスキーがインストールされていない状態でシステムが起動すると、デフォルトの制限付きのセッション使用ライセンスと機能ライセンス一式がインストールされます。次の Exec モードコマンドは、ライセンス情報を一覧表示します。

`[local]host_name#` **show license information**

重要 ライセンスキーがインストールされていない場合、PDSN、HA、GGSN、および L2TP LNS のセッション使用ライセンスは 10,000 セッションに制限されます。

セッション使用とライセンス使用のライセンス

セッション使用および機能の使用ライセンスは、セッション制限を制御し、システム内の特別な機能を有効にするソフトウェアメカニズムです。これらの電子ライセンスは、システムが電源オンになるか再起動されるたびに、システムソフトウェアの一部としてロードされるシステム コンフィギュレーション ファイルに保存されます。

- セッション使用ライセンスは、システムがサービスタイプごとにサポートできる同時セッション数を制限し、必要に応じて取得されます。これにより、キャリアは利用しているものに対してのみ支払いを行い、サブスクライバベースの増加に応じてキャパシティを簡単に増やすことができます。
- 機能使用ライセンスは、システム内で特定の機能を有効にし、システムでサポートされているセッションの合計数に基づいて配布されます。

新しいライセンスキーのインストール

新しいライセンスキーをインストールするには、次の手順を使用します。

キーのカットアンドペースト

ライセンスのコピーがある場合は、次の設定を使用して、ライセンスキー部分だけをカットアンドペーストします。

ステップ 1 Exec モードで、次のように入力します。

```
configure
  license key license
  exit
```

license はライセンスキー文字列です。ライセンスは、大文字と小文字が区別される 1 ～ 1023 文字の英数字文字列にすることができます。次の例に示すように、ライセンスキーをコピーします（「\(二重引用符)」を含む）。これは機能ライセンスではありませんのでご注意ください。

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\
STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|
LSP=000000|LSH=000000|LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|
FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/
XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

ステップ 2 Exec モードプロンプトで次のコマンドを入力して、入力したライセンスキーが受け入れられたことを確認します。

```
[local]host_name# show license key
```

新しいライセンスキーが表示されます。表示されない場合は、グローバル コンフィギュレーション モードに戻り、**license key** コマンドを使用してキーを再入力します。

重要 無効なライセンスは受け入れられません。無効なライセンスキーを設定しようとすると、**license key** コマンドの出力に障害エラーが表示されます。**-force** オプションを使用して無効なライセンスキーをインストールすると、ライセンスは30日間の猶予期間に入ります。StarOSは、猶予期間中に毎日のsyslogエラーメッセージとSNMPトラップを生成します。**show license information** コマンドの出力には、「License State」が「Not Valid」として示されます。

ステップ3 次のコマンドを入力して、ライセンスキーが正しい機能を有効にしていることを確認します。

```
[local]host_name# show license information
```

すべてのライセンスキーと、有効になっている新しいセッションのキャパシティまたは機能が表示されます。新しいキーで有効になっている機能またはセッションのキャパシティが正しくない場合は、サービス担当者にお問い合わせください。

ステップ4 「設定の確認と保存」の章の説明に従って、設定を保存します。

注意 新しいライセンスキー設定を現在のCLIコンフィギュレーションファイルに保存しないと、システムがリロードされた際に、ライセンスキーによって有効にされた新しい機能が失われます。

ライセンスキーをコンフィギュレーション ファイルに追加

ライセンスキーは、新規または既存のコンフィギュレーション ファイルに追加できます。

重要 ライセンスキー情報は、CLI設定の一部として維持されます。キーがインストールまたは更新されるたびに、コンフィギュレーション ファイルを再保存する必要があります。

ステップ1 新しいライセンスキーコマンドをコピーするコンフィギュレーションファイルを開きます。

ステップ2 例に示すように、「\（二重引用符）」を含むライセンスをコピーします。これは機能ライセンスではありませんのでご注意ください。

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\STCB21M82003R80411A4|
DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|LSP=000000|LSH=000000|
LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|
SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

ステップ3 ライセンスキーを設定に貼り付けます。

重要 コンフィギュレーション ファイルの先頭にライセンスキー情報を貼り付けて、システムがコンテキストを設定する前に期待される容量と機能を確保するようにします。

ステップ4 「設定の確認と保存」の章の説明に従って、設定を保存します。

ライセンスの期限切れの動作

ライセンスの有効期限が切れると、30日の猶予期間が設定され、ライセンス済みセッション使用および機能使用ライセンスの通常の使用が可能になります。これにより、サービスを中断せずに新しいライセンスを取得できます。

次の Exec モードのコマンドは、猶予期間の有効期限に設定された日付を含むライセンス情報を一覧表示します。

show license information

ライセンスキーの要求

システムのライセンスキーは、シスコのアカウント担当者から入手できます。ライセンスキーを生成するには、特定の情報が必要です。

- SO または発注書情報
- 必要なセッションキャパシティ
- 必要な機能

ライセンス情報の表示

ライセンスの詳細を表示するには、Exec モードで次のコマンドを入力します。

`[local]host_name#` **show license information [full | key [full]]**

ライセンスキーの削除

セッションおよび機能使用ライセンスキーを設定から削除するには、次の手順を実行します。セキュリティ管理者または管理者である必要があります。

```
configure
  no license key
  exit
show license key
```

このコマンドの出力には、「No license key installed」と表示されています。

ローカルユーザー管理アカウントの管理

設定ファイルを介して設定されたコンテキストレベルの管理アカウントとは異なり、ローカルユーザー管理アカウントの情報は、フラッシュメモリ内の別のファイルに保持され、ソフトウェアの共有設定タスク（SCT）によって管理されます。ローカルユーザーアカウントはANSI T1.276-2003 に準拠するように設計されているため、システムにはこれらのタイプの管理ユーザーアカウントを管理するためのさまざまなメカニズムが備わっています。

詳細については、コンソールの AAA ベースの認証を無効化およびコンソール/VTY 回線でのローカルユーザーログインの制限を参照してください。

ローカルユーザーパスワードのプロパティの設定

ローカルユーザーアカウントのパスワードプロパティはグローバルに設定され、すべてのローカルユーザーアカウントに適用されます。システムでは、次のパスワードプロパティの設定がサポートされています。

- **Complexity**：パスワードの複雑さは、ANSI T1.276-2003 に強制的に準拠させることができます。
- **History length**：システムで追跡する必要がある以前のパスワードバージョンの数。
- **Maximum age**：ユーザーが同じパスワードを使用できる期間。
- **Minimum number of characters to change**：リセット時にパスワードで変更する必要がある文字数。
- **Minimum change interval**：ユーザーがパスワードを変更できる頻度。
- **Minimum length**：有効なパスワードに含める必要がある最小文字数。
- **Expiry warning**：パスワードの有効期限の警告間隔（日数）。
- **Auto-generate**：パスワードの長さを指定するオプションを使用して、自動的にパスワードを生成します。

上記パラメータそれぞれの詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user password** コマンドを参照してください。

ローカルユーザーのアカウント管理プロパティの設定

ローカルユーザーのアカウント管理には、アカウントのロックアウトとユーザーの一時停止が含まれています。

ローカルユーザーアカウントのロックアウト

ローカルユーザーアカウントは、次の理由で管理上ロックすることができます。

- **ログインの失敗**：設定された最大ログイン失敗のしきい値に達しました。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user max-failed-logins** コマンドを参照してください。
- **パスワードのエージング**：設定された最大パスワードの有効期限に達しました。詳細については、『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user password** コマンドを参照してください。

ロックアウトされたアカウントは、設定されたロックアウト時間に達するまで（『*Command Line Interface Reference*』の「*Global Configuration Mode Commands*」の章で説明されている **local-user lockout-time** コマンドを参照）、またはセキュリティ管理者がロックアウトをクリアするまで（『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章で説明されている **clear local-user** コマンドを参照）、ユーザーはアクセスできません。

重要 ローカルユーザーの管理ユーザーアカウントは、ロックアウトを適用または拒否するように設定できます。詳細については、『*Command Line Interface Reference* 』の「*Global Configuration Mode Commands*」の章で説明されている **local-user username** コマンドを参照してください。

ローカルユーザーアカウントの一時停止

ローカルユーザーアカウントは、次のように一時停止することができます。

```
configure
  suspend local-user name
```

次のように入力して、一時停止を削除できます。

```
configure
  no suspend local-user name
```

ローカルユーザーパスワードの変更

ローカルユーザーの管理ユーザーは、Exec モードで **password change** コマンドを使用してパスワードを変更できます。ユーザーは、現在のパスワードと新しいパスワードを入力するように求められます。

セキュリティ管理者は、Exec モードで root プロンプトから次のコマンドを入力して、ローカルユーザーのパスワードをリセットできます。

```
[local]host_name# password change username name
```

name は、パスワードを変更するローカルユーザーアカウントの名前です。セキュリティ管理者がローカルユーザーのパスワードをリセットすると、ユーザーは次回ログイン時にパスワードを変更するように求められます。

新しいパスワードは、システムに設定されているパスワードプロパティに従う必要があります。

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。