Cisco SD-Access のプロビジョニングのワークフローとトラブルシューティング ガイド

改訂：2020 年 10 月 16 日

サービス プロビジョニング フレームワーク

このガイドでは、Cisco DNA Center のデバイスとファブリックのプロビジョニングに関する手順について説明します。このガイドでは、重要な API と問題のトラブルシューティング方法についても説明します。

サービスプロビジョニングフレームワーク（SPF）は、RESTful インターフェイスを介して、ネットワークサービスおよびネットワークデバイスのポリシーのプロビジョニングと設定を容易にします。SPF は、オーケストレーション エンジンを使用して内部実行フローを実装し、顧客向けのサービスからリソース向けのサービス（CFS から RFS）へのマッピングを実行します。

SPF は、Maglev クラスタ上で実行される一連のマイクロサービスです。

SPF の用語とコアサービス

SPF では、次の用語が使用されます。

- 顧客向けサービス：顧客から見たサービスの商業的な見え方。必要な技術的機能を提供する一連のリソース向けサービスをグループ化します。

  VPN は、顧客向けサービスの一例です。

- リソース向けサービス：間接的に製品の一部ですが、顧客には見えません。これは、1 つ以上の顧客向けサービスをサポートするために存在するものです。

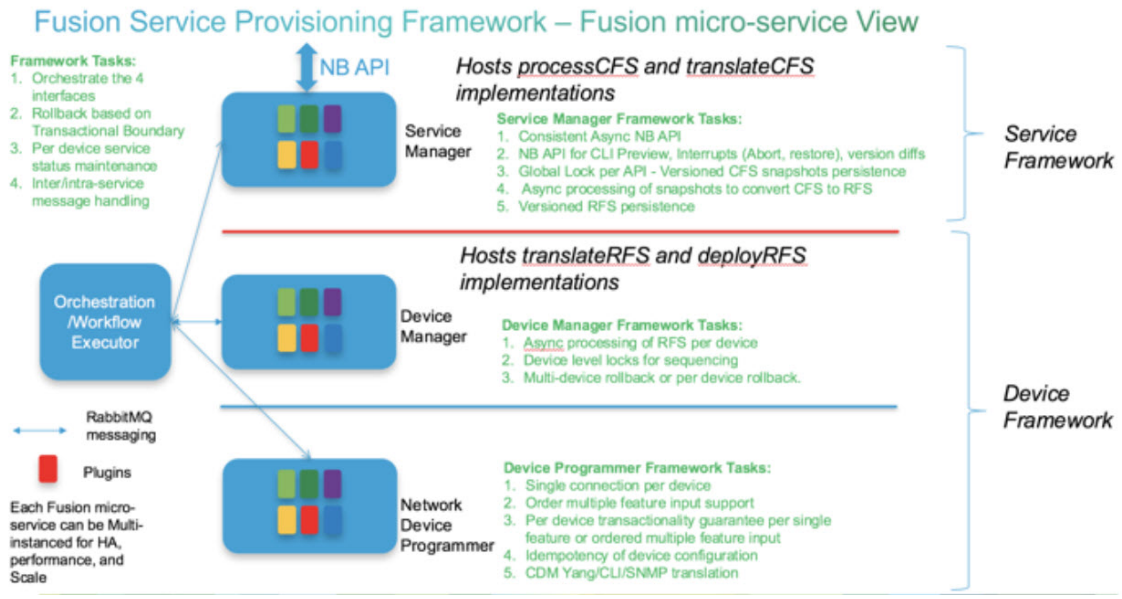
  VPN の例では、サポートする BGP が必要です。顧客は BGP を購入しておらず、BGP が稼働していることすら認識していないと思われます。つまり、BGP はリソース向けサービスの一例です。

SPF には次のコアサービスが含まれています。

- *spf-service-manager-service*：プロビジョニング ワークフローを開始し管理します。
- *spf-device-manager-service*：デバイスレベルの RFS 仕様とデバイスレベルのロックを生成し、設定の展開を開始します。
- *apic-em-network-programmer-service*：RFS を CLI コマンドに変換し、デバイスにプッシュします。
- *task-service*：プロビジョニング ワークフロー タスクを作成して維持します。
- *orchestration-engine-service*：プロビジョニング ワークフローを実行します。
- *rabbitmq-service*：プロビジョニング ワークフロー中にさまざまなサービス間の非同期メッセージングを処理します。

SPF アーキテクチャ図

次の図は、サービスレイヤおよびデバイス フレームワーク レイヤを含む自動化コンポーネントのマイクロサービス ビューを示しています。



SPF 診断

SPF 診断 API

SPF 診断APIは、SPF ノースバウンドインターフェイスの一部として使用でき、プロビジョニング要求の一部として実行されるすべてのステップを収集します。SPF 診断 API は taskid または workflowid を使用し、プロビジョニングの問題をデバッグするために必要な詳細を提供します。

```
https://<server-ip>/api/v2/data/spf-diagnostics/workflowinfo?taskid=d54b03b6-71bd-46ae-84b0-50412eab...
```

```
https://<server-ip>/api/v2/data/spf-diagnostics/workflowinfo?workflowid=f5a66f1e-b680-46ce-b362-860d...
```

タスク API

タスク API は、SPF プロビジョニング操作用に作成されたタスクを取得します。

```
https://<server-ip>/api/v1/task?serviceType=SPFService&progress=TASK_MODIFY_PUT&isError=true&sortB...
```

rabbitmq Console への移動

```
#First enable rabbitmq management
magctl service attach rabbitmq
rabbitmq-plugins enable rabbitmq_management

# On the root expose port. That will return to you random port
magctl service expose rabbitmq 15672

# Go rabbit console.
http://<ip>:<port that you just got though expose>/


# Credentials
```

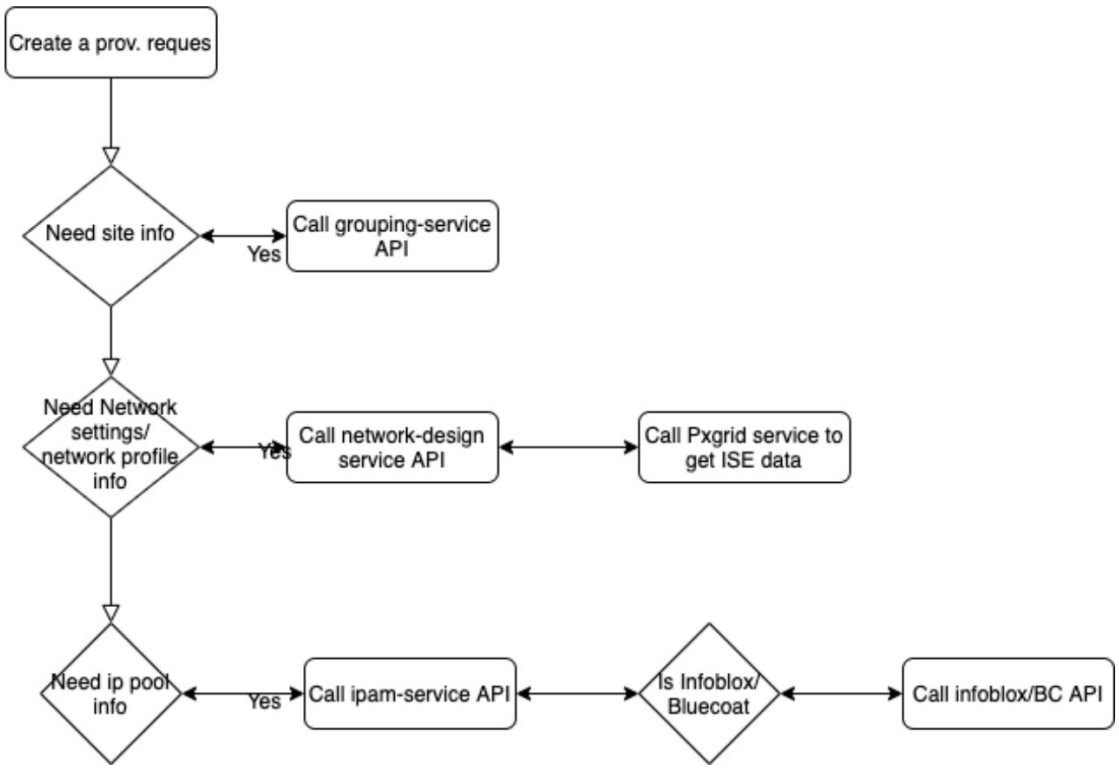
```
To get password, on the root, run:
$ magctl service exec rabbitmq "cat /vault/rabbitmq-appuser"
Defaulting container name to rabbitmq.
Use 'kubectl describe pod/rabbitmq-0' to see all of the containers in this pod.
<password>


Username will always be: appuser
```

プロビジョニング要求の生成

次の図は、Cisco DNA Center の GUI からのプロビジョニング要求の作成に関連するさまざまなサービスを示しています。これらのサービスは、デバイスプロビジョニング およびファブリック プロビジョニング ワークフローで使用されます。

サービスが使用できない場合、プロビジョニング要求に影響します。



プロビジョニングのワークフロー

プロビジョニング操作には、次の主要な手順が含まれます。プロビジョニング操作でスイッチに設定をプッシュする必要がない場合、2 番目の手順はスキップされます。

1. CFS（UI/API 入力）を RFS/BCS（プッシュ設定）に変換します。
2. ターゲットデバイスに RFS を展開します。

手順 1：CFS の BCS への変換

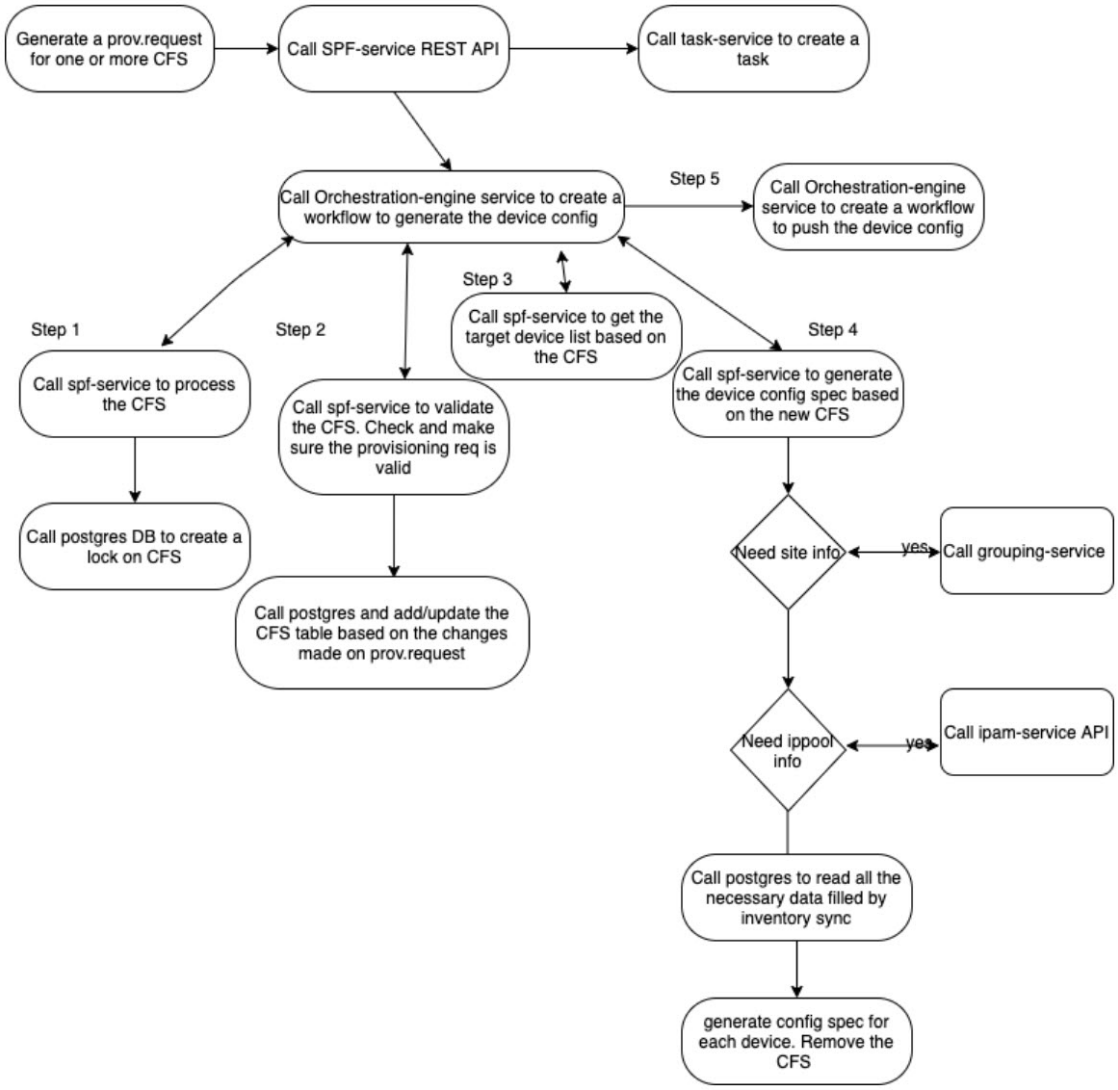
CFS をデバイスベースの RFS に変換し、デバイスにプッシュできるようにするには、次の手順を実行します。

1. *CFS* プリプロセッサ：サービスプロビジョニング要求のスナップショットを作成し、CFS のロックを作成して、現在の変更が完了するまで後続のプロビジョニング操作を待機するようにします。
2. *CFS* 検証：サービスプロビジョニング要求を検証し、すべての前提条件が満たされていることを確認します。

   たとえば、デバイスのプロビジョニング中に、検証によってデバイスがインベントリに存在し、アクティブな LAN 自動化の一部ではないことを確認します。
3. *CFS* プロセッサ：CFS モデルのスナップショットを作成し、対応する CFS およびシリアル化されたスナップショットのデータベーステーブルに保存します。スナップショットはグローバルロック下で取得されるため、競合状態なしで真のソースのスナップショットが 1 つだけ生成されます。ロックは、スナップショットがデータベースに保持された直後に放棄されます。

   デバイスのプロビジョニング時に、Deviceinfo の ID を使用して CustomerFacingService テーブルにエントリが作成されます。
4. *CFS* ターゲットリゾルバ：プロビジョニングするターゲットデバイス ID を検索します。複数のデバイスを選択して VN 更新をプロビジョニングまたは実行する場合、この手順によって関連デバイスのリストが決定されます。
5. *CFS* トランスレータ：シリアル化されたスナップショットのデータベーステーブルに保存されている日付を読み取り、ユーザ入力を変換してデバイス固有の RFS をプロビジョニングします。この情報は spf-device manager に送信されます。

   この段階で、Cisco DNA Center はインベントリ対応データベーステーブルから対応する RFS エントリを読み取ります。次に、Cisco DNA Center は CFS で行った変更で RFS オブジェクトを更新します。



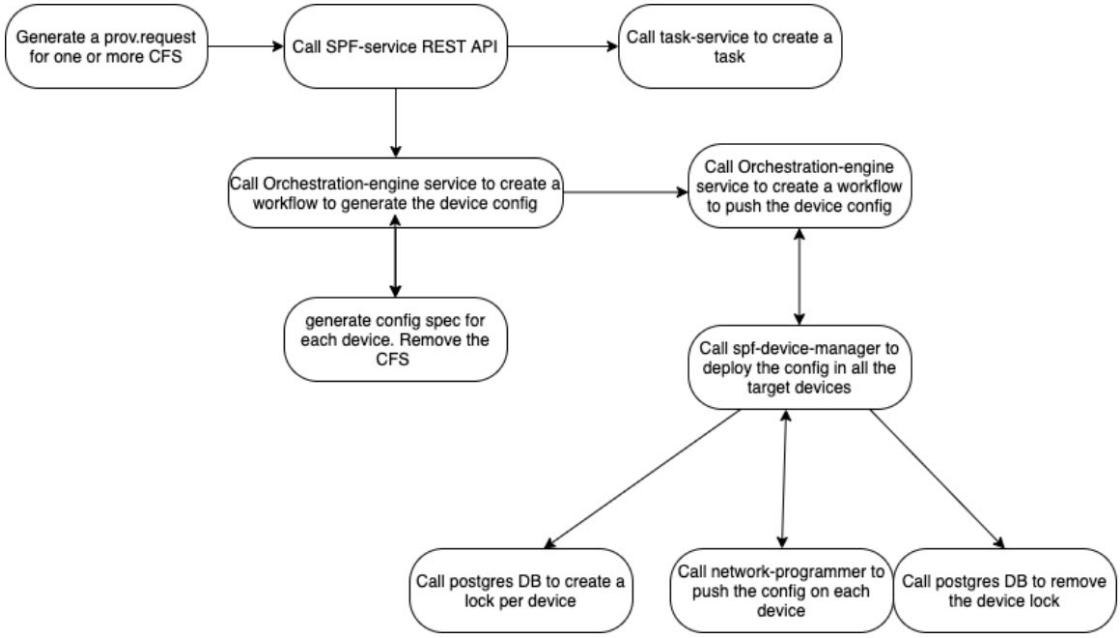
手順 2：ターゲットデバイスへの設定の展開

RFS をターゲットデバイスに展開するには、次の手順を実行します。

1. *RFS* トランスレータ：bulkconfigspec を読み取り、デバイスレベルの RFS オブジェクトを作成してデバイスにプッシュします。

   この段階で、Cisco DNA Center ではすべてのプロビジョニング関連のデータベーステーブルにデバイスレベルのデータベースロックが作成されます。

2. *RFS* デプロイヤ：特定の RFS の機能名（デバイスパック作成者名）の入力を含む、network programmer に送信するデバイスごとの機能ドキュメントを作成します。RFS デプロイヤは、定義されたロールバックルール（別のデバイスへの設定が失敗した場合の正常に設定されたデバイスのロールバックなど）に基づいて、複数のデバイス間でのロールバックを処理します。RFS デプロイヤは、デバイスごとの展開の結果も更新します。



プロビジョニングに関連する他のサービス

プロビジョニングには、次のサービスが含まれます。

- *network-design-service*：ネットワーク設定とデバイスクレデンシャル CRUD の操作を処理します。ネットワークプロファイルを作成し、サイトへの関連付けを行います。
- *orchestration-engine-service*：プロビジョニング ワークフローを実行します。
- *identity-manager-pxgrid-service*：Cisco ISE が AAA サーバとして設定されている場合、このサービスはすべての Cisco ISE サーバ情報を取得します。
- *apic-em-inventory-manager-service*：デバイスが Cisco DNA Center のインベントリで検出され、それらの同期ステータスが Partial Collection Failure ではないことを確認します。

  CFS から RFS への変換フェーズ中に、プロビジョニングアプリはデータベーステーブルから値を読み取り、それらのモデルを更新してプロビジョニングします。
- *grouping-service*：サイトの作成と変更を処理します。サイトの欠落でデバイスのプロビジョニングが失敗する場合は、grouping-service を確認します。

  いずれかのサイトでネットワーク設定が欠落している、または継承が正しくない場合は、grouping-service を確認し、サイト更新通知が正しく機能していることを確認します。
- *template-programmer-service*：プロビジョニング ワークフローにアタッチされているテンプレートを実行します。

重要な API

次のAPIが重要です。

- *api/v2/data/customer-facing-service/<CFSType>*

  たとえば、**api / v2 / data / customer-faceing-service / deviceinfo** はすべてのデバイス関連データを取得します。

- *api/v1/siteprofile*

  たとえば、**api / v1 / siteprofile？namespace = authentication** はすべての認証プロファイルをリストします。

- *api/v2/ippool/group?siteId=<siteId>*

  選択したサイトで予約されているすべての IP プールを一覧表示します。

- *api/v1/commonsetting/*

  たとえば、**api / v1 / commonsetting / global** は、すべてのグローバルネットワーク設定をリストします。

表 *1 : CFS* と *RFS* のマッピング

サービス名	CFS サービスタイプ	CFS テーブル	RFS テーブル
Device Provisioning	DeviceInfo	DeviceInfo Networkwidesettingscfs	Networkdevice AAANeSettings
ファブリックの作成 ファブリックトランジットの作成	ConnectivityDomain	connectivitydomain Virtualnetwork	—
ファブリックへのサイトの追加 仮想ネットワークの認証テンプレートの更新	ConnectivityDomain	Connectivitydomain Virtualnetwork authenticationprofile	Vrf Dot1xNeSettings
ファブリックへのエッジの追加	ConnectivityDomain	Connecitivtydomain Deviceinfo	Lispcomponent Networkvlan DhcpServerSettings LispItrMapCacheEntry
ファブリックへのボーダーの追加	ConnectivityDomain	Connecitivtydomain Deviceinfo	Lispcomponent Networkvlan DhcpServerSettings BgpProcessSettings RoutePolicyMap BgpNeighborInfo

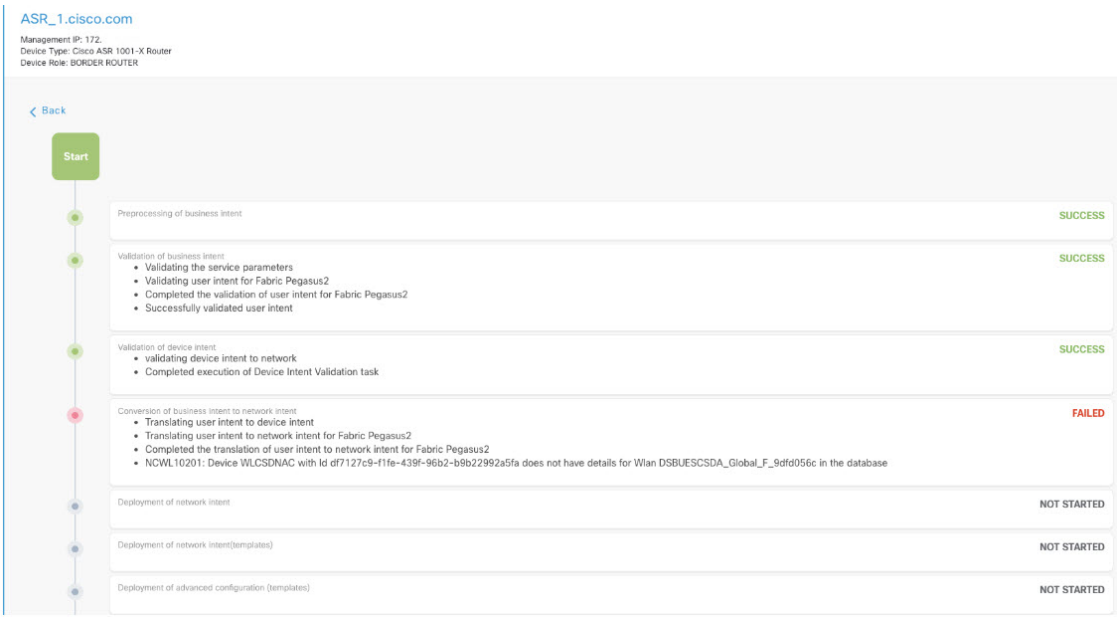
サービス名	CFS サービスタイプ	CFS テーブル	RFS テーブル
マルチキャストの有効化	ConnectivityDomain	Connecitivtydomain Deviceinfo	igmpNrSettings MsdpNrSettings IpV4PimNrSettings IpV4MulticastNrSettings IpV4PimNrSettings
仮想ネットワークの IP プールの追加	ConnectivityDomain	Connecitivtydomain virtualnetwork	Segment Vrf protocolendpoint
スタティックポートの割り当て	ConnectivityDomain	Connecitivtydomain Deviceinfo deviceinterfaceinfo	protocolendpoint

トラブルシューティング

デバイスプロビジョニングの失敗は珍しくありません。障害をトラブルシューティングするには、次の手順を実行します。

手順

ステップ 1 問題のあるデバイスの [Provisioning Status] で、[See details] をクリックして、最近のプロビジョニングタスクとステータスのリストを表示します。最新のプロビジョニングタスクについて [See details] をクリックします。
次のようなウィンドウが表示されます。



ステップ 2 次のプロビジョニングタスクのいずれかのステータスが FAILED の場合は、spf-service-manager ログファイルを確認します。

- ビジネスインテントの前処理
- ビジネスインテントの検証
- デバイスインテントの検証
- ビジネスインテントをネットワークインテントに変換する

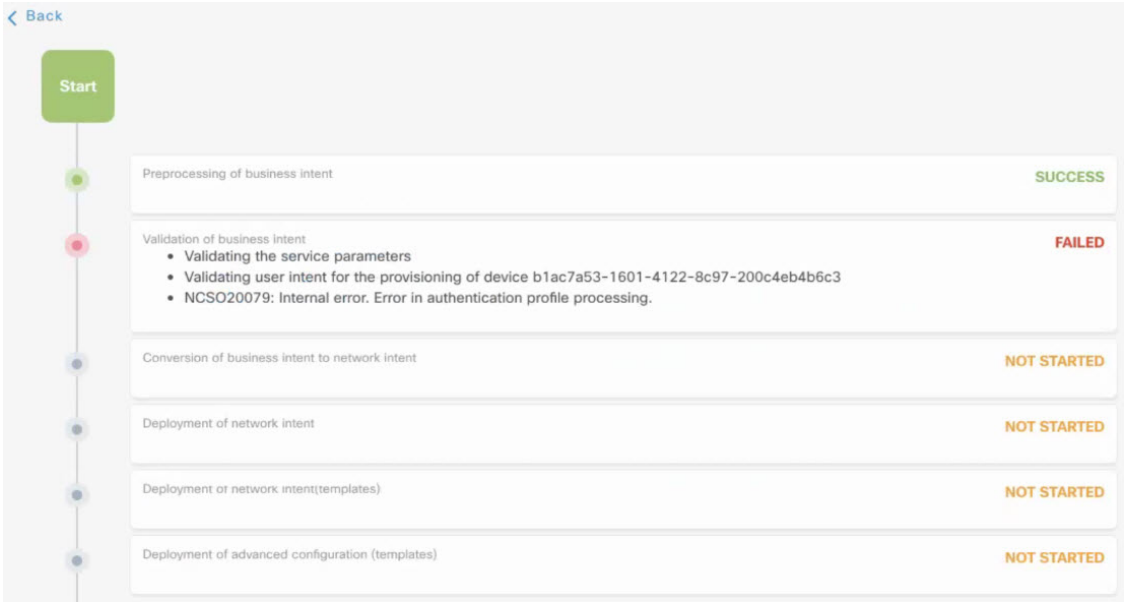
a) spf-service-manager ログファイルで、「Creating task:」というキーワードを検索します。
最新のプロビジョニング操作のタスク ID は、「Creating task:」キーワードの後に出力されます。次に例を示します。

```
Creating task:d20a92d1-c0fd-47bc-8db7-6417a362f5f7
```

b) 失敗したプロビジョニングのステップを確認するには、ログファイルで前述のタスク ID を検索します。失敗したステップには、エラーの完全な例外スタックトレースが含まれています。

ステップ 3 検証フェーズでプロビジョニングが失敗した場合は、プロビジョニング要求の一部として返される JSON 要求を調べます。JSON 要求には「Service Request」の文字列とそれに続くタスク ID が含まれています。たとえば、次の文字列は、AAA サーバがプロビジョニング要求または CFS テーブルにないことを示します。

```
Device provisioning Failed with the following error: Error in authentication profile processing.
```



a) AAA サーバ設定がプロビジョニング要求にない場合、JSON 要求は次の例のようになります。[Design] ウィンドウで、問題のあるデバイスが存在するサイトの AAA 設定を確認し、AAA サーバエントリが欠落していないか確認します。

```
JSON Request:
task id:8890786e-9fbe-430e-b670-70d306ef6d09;
service request:
[{deviceInterfaceInfo=[{interfaceId=<ID>, role=LAN,
segment=[{idRef=2e1d6385-1abf-48a5-a655-574f8ce10ead}], connectedDeviceType=USER_DEVICE,
description=pls06-lnl-001.cisco.com,
authenticationProfile={idRef=6ecab335-abe6-4147-b7d5-3eb5bf4a2f63}}],
displayName=f5ba231e[da4ba8eb-0b7e-4713-ade0-6db41089cac7], cfsChangeInfo=[],
createTime=1572718371294,
deployed=false, id=<ID>, instanceId=92916868, instanceVersion=23, isStale=false,
lastUpdateTime=1573410346832, name=pls06-sd-sw1.cisco.com,
namespace=50e14b21-e176-4073-944b-f85abb509dc5,
networkDeviceId=<ID>, networkWideSettings={id=<ID>,
instanceId=82632579, displayName=0, instanceVersion=23, aaa=[], dns=[{id=<ID>,
domainName=cisco.com, ip={id=<ID>, address=<IP>},
secondaryIp={id=<ID>, address=<IP>}}], ldap=[], nativeVlan=[],
netflow=[], ntp=[{id=<ID> ipAddress={id=<ID>,
address=<IP>}}, {id=<ID>, ipAddress={id=<ID>,
address=<IP>}}], snmp=[], syslogs=[]}, provisioningState=DEFINED, resourceVersion=23,
roles=[EDGENODE],
siteId=001812a1-e718-479e-8c43-6b007b18ffca, transitNetworks=[], type=DeviceInfo, wlan=[],
otherDevice=[],
deviceInterfaceInfoContainer={idRef=63194545-00ca-4796-a86c-bba4b1c1f8bf}}];
```

ステップ 4 ビジネスインテントからネットワークインテントへの変換フェーズでプロビジョニングが失敗した場合は、タスク ID に基づいて例外スタックトレースを確認します。次のエラーを探します。

```
Error while doing async CFS translation.
```

ログメッセージの例：

```
at java.base/java.lang.Thread.run(Thread.java:834)
Caused by: java.lang.NullPointerException
at com.cisco.ef.lan.rfs.helper.SegmentRfsGenerator.generateSVIOnDevice(SegmentRfsGenerator.java:398)
```

```
at com.cisco.ef.lan.rfs.helper.LanRfsGenerator.handleSegmentCreateForDevice(LanRfsGenerator.java:2309)
at
com.cisco.ef.lan.rfs.helper.LanRfsGenerator.handleOptimizedVNandSegmentOperationForDevice(LanRfsGenerator.java:536)
at com.cisco.ef.lan.rfs.helper.LanRfsGenerator.generateOptimizedRFSForDevice(LanRfsGenerator.java:844)
at com.cisco.ef.lan.rfs.helper.LanRfsAsyncGenerator.call_aroundBody0(LanRfsAsyncGenerator.java:40)
at
com.cisco.ef.lan.rfs.helper.LanRfsAsyncGenerator.call_aroundBody1$advice(LanRfsAsyncGenerator.java:171)
at com.cisco.ef.lan.rfs.helper.LanRfsAsyncGenerator.call(LanRfsAsyncGenerator.java:1)
at com.cisco.ef.lan.rfs.helper.LanRfsAsyncGenerator.call(LanRfsAsyncGenerator.java:14)
at java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)

2020-01-13 21:53:32,841 |  ERROR | Translate_Cfs_Thread_26   |  | c.cisco.dnac.error.log.ErrorLogger
 |
NCSO10008: Error in generating RFS due to internal error. generateSVIOnDevice: BCS doesnt have
vrfNames : VN_TEST_ |
```

a) 上記のエラーメッセージに基づくと、Cisco DNA Center が RFS テーブルデータに基づいて設定を作成しようとするときには「VN_TEST_」という名前のVRFが必要ですが、これが不足しています。データベーステーブル VRF の定義を確認します。

```
campus=# \d vrf;
                        Table "public.vrf"
       Column       |          Type           |     Modifiers
--------------------+-------------------------+--------------------
 instance_version   | integer                 | not null
 id                 | bigint                  | not null
 instanceuuid       | character varying(255)  |
 displayname        | character varying(255)  |
 name               | character varying(255)  | not null
 description        | character varying(4000) |
 owningentityid     | character varying(255)  | not null
 routedistinguisher | character varying(255)  |
 isrdauto           | boolean                 | not null default 0
 vpnid              | character varying(255)  |
 networkresource_id | bigint                  |
 deploypending      | integer                 |
 authentityid       | bigint                  |
 authentityclass    | integer                 |
 instancetenantid   | integer                 |
 instanceorigin     | integer                 |
 tenantintsegment   | integer                 | default 0
 tenantlongsegment  | bigint                  | default 0
Indexes:
```

b) owningentityid フィールドは、テーブルが networkdevice テーブル（メインのインベントリテーブル）を参照することを示しています。次の SQL クエリを入力して、VRF テーブルに VRF エントリがあるか確認します。

最初のクエリ：

```
select id from networkdevice where managementipaddress='x.x.x.x';
```

2 番目のクエリは、最初のクエリ応答から ID を渡します。

```
select * from vrf where owningentityid like '<id>%';
```

c) VRF エントリがない場合は、インベントリに問題があります。デバイスのインベントリの状態を確認します。デバイスのステータスが [Partial Collection Failure]（PCF）の場合は、問題を修正し、デバイスのステータスが [Managed] に変わることを確認し、上記の SQL クエリを再実行します。

ステップ 5 次のいずれかのプロビジョニングタスクのステータスが FAILED の場合は、spf-device-manager および network-programmer のログファイルを確認します。

- ネットワークインテントの導入
- ネットワークインテント（テンプレート）の導入

次に例を示します。

```
Time:November 30, 2019 11:08 AM
Task:N/A
Status:FAILED
Error:Unable to push CLI 'no propagate sgt' to device <IP> using protocol ssh2

Time:November 30, 2019 12:37 PM
Task:N/A
Status:FAILED
Error:Enable password is incorrect for device <IP>
```

a) 次の SQL クエリを入力して、タスクが失敗している正確な CLI コマンドを取得します。

```
select error_message from deviceconfigstatus where deviceid in(select instanceuuid from
networkdevice
where managementipaddress like 'x.x.x.x');
```

b) spf-service ログファイルからタスク ID を取得し、spf-device-manager ログファイルでそのタスク ID を検索します。タスク ID は、完全なステータスメッセージと例外スタックトレースを提供します。

次に例を示します。

```
UnmanagedDeviceConfigStatus[managedDeviceId=<ID>,cfsVersion=8,createTime=1586331074235,deploymentErrors=
Unable to push configuration to device ip
<IP>.,deviceId=<ID>,error=DeviceConfigError[{},com.cisco.nm.pal.client.PALException:
<palError><deviceId>815821</deviceId><code>HANDLER_ERROR</code><version>3.3.31</version><itemLocation>
[xmp_push_to_device_utility]configCollectors/ConfigCollectors.xpa/pushConfig/pushCli.par@Rule0/HandlerChain/devkt:
170324204049919.try.kalmar:190902142423406.steps.kalmar:190902142431311.try.Handler1</itemLocation><message>
Failed to match expected device output due to expect timeout, current expect timeout 300000ms,
 expect time 300000ms,
minimal matching length 0.
| 6286 | Current output : do ap name KRK-LAP1 location "Global/Cisco Poland/KrakC3w/Budynek
C/Pietro-2"
| 6287 | Site2-FIAB(config)#
| 6288 | Current expects : do ap name KRK-LAP1 location "Global/Cisco Poland/Kraków/Budynek
C/Pietro-2"
| 6289 | </message><handlerCode>ERROR_TIMEOUT</handlerCode><errorMatchingOutput><![CDATA[do ap
 name KRK-LAP1 location
"Global/Cisco Poland/KrakC3w/Budynek C/Pietro-2"
| 6290 |
Site2-FIAB(config)#]]></errorMatchingOutput><errorName>echoExpectError</errorName><lastCommandSent>do
 ap name
KRK-LAP1 location "Global/Cisco Poland/Kraków/Budynek
C/Pietro-2"</lastCommandSent><lastCommandSentIndex>2</lastCommandSentIndex>
<fullTranscript><![CDATA[config t
| 6291 |
| 6292 | Enter configuration commands, one per line. End with CNTL/Z.
| 6293 | Site2-FIAB(config)#do ap name KRK-LAP1 location "Global/Cisco Poland/Kraków/Budynek
C/Pietro-2"
| 6294 | ]]></fullTranscript><sessionDetails><![CDATA[Run ID 12951. CLI session 722 on device
815821. Session duration 300200ms.
Session response duration never. SessionSourceNewlyCreated.
SessionProgressCommandExecution.]]></sessionDetails><credential location= "DCMInventoryProvider"
 name="CLI_ADDRESS">100.127.0.102</credential><credential location="XdeIOSExt.def[USER_DEFINED]"
 name= "CLI_CONFIG_PROMPT_SEARCH_EXPRESION">(.{0,10})(.*)(\S)\s*$</credential><credential
```

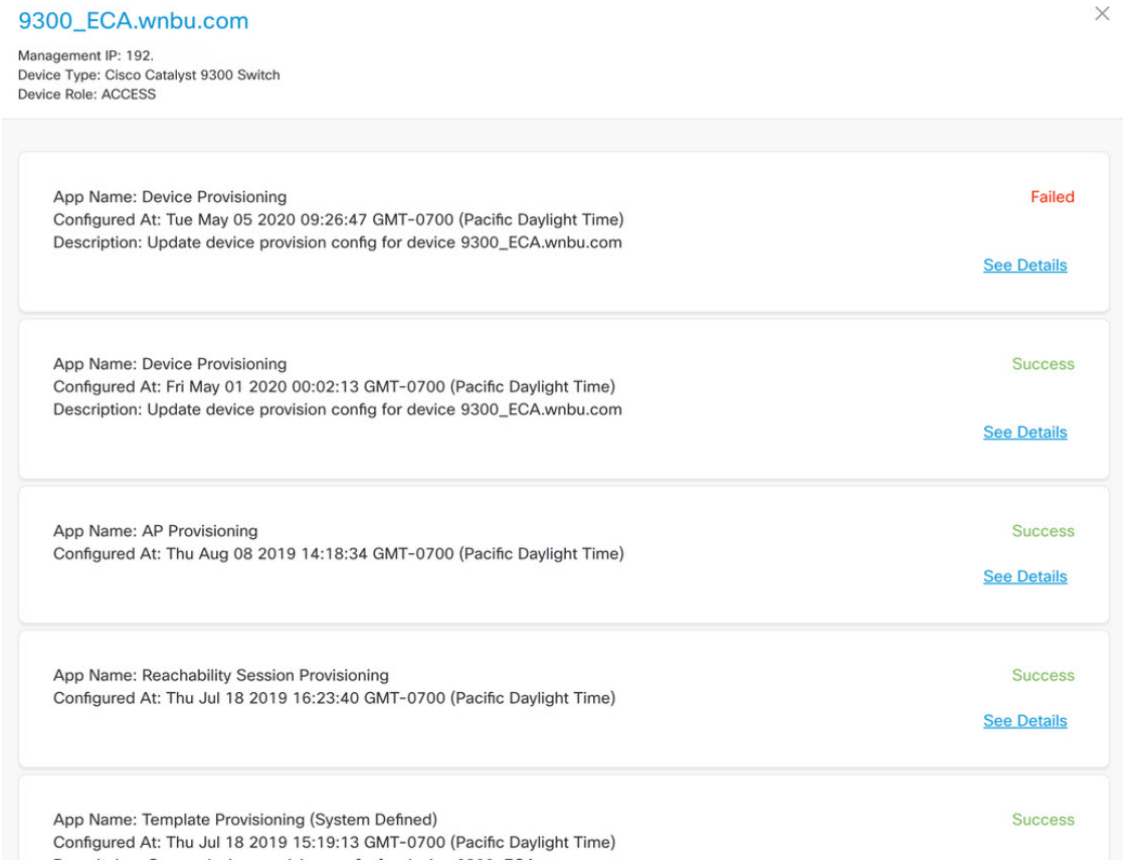
```
location="DCMInventoryProvider" name= "CLI_CONNECTION_TIME">120</credential><credential
location="DCMInventoryProvider" name="CLI_ENABLE_CONDITION">ALWAYS</credential>
<credential location="DCMInventoryProvider" name="CLI_ENABLE_PASSWORD">****</credential><credential
 location="DCMInventoryProvider"  name="CLI_LOGIN_PASSWORD">****</credential><credential
location="DCMInventoryProvider" name="CLI_LOGIN_USERNAME">netadmin</credential>
<credential location="DCMInventoryProvider"
name="CLI_PASSWORD_EXPECT">assword[:\s]*\z</credential><credential location="DCMInventoryProvider"
  name="CLI_PORT">22</credential><credential location="XdeIOS_ngwc.def[USER_DEFINED]"
name="CLI_PROMPT_EXPECT">[\(\)\d\w\{\}\.]\s?[#>\$]\s*\z</credential>
<credential location="SetCredential" name="CLI_SINGLE_BUFFER_SENDING">TRUE</credential><credential
 location="XdeIOS_ngwc.def[USER_DEFINED]"
name="CLI_SINGLE_BYTE_READING">true</credential><credential location="DCMInventoryProvider"
name="CLI_TRANSPORT">ssh2</credential><credential location= "DCMInventoryProvider"
name="CLI_USERNAME_EXPECT">ogin[:\s]*\z
| 6295 | Name[:\s]*\z
| 6296 | name[:\s]*\z
| 6297 | User[:\s]*\z</credential><credential
name="DEVICE_CONNECTION_TIMEOUT"><null></credential><credential location= "DCMInventoryProvider"
 name="DEVICE_FAILSAFE_TIMEOUT">3600000</credential><credential
name="DEVICE_ID">815821</credential><credential name= "DEVICE_NAME">815821</credential><credential
 location="DCMInventoryProvider" name="DEVICE_TIMEOUT">300000</credential><credential location=
 "DCMInventoryProvider" name="software">IOS</credential><credential location="DCMInventoryProvider"
 name="variant">XE</credential><credential location= "DCMInventoryProvider"
name="version">16.12.1s</credential></palError>,NP_300,Unable to push configuration to device
ip
```

集約済みのプロビジョニングステータス

各デバイスのプロビジョニングステータスは、さまざまなアプリケーション間で集約されます。

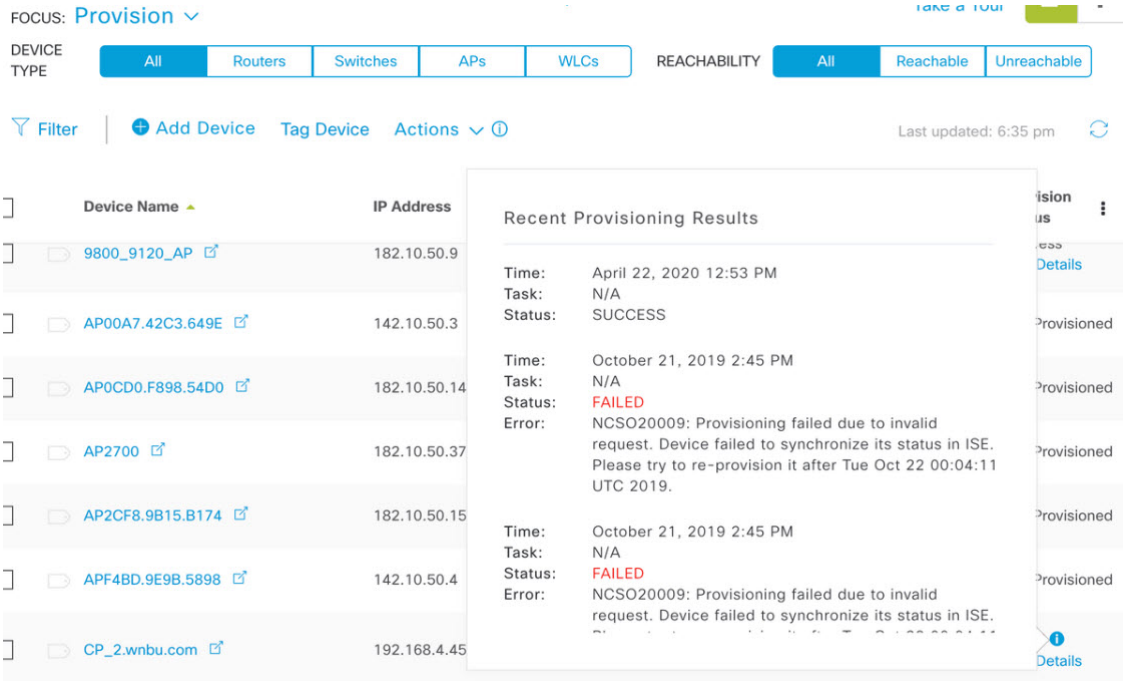
たとえば、次のプロビジョニング操作では、集約ステータスは *Failed* になります。

- 最新のデバイス プロビジョニング ステータス：Failed
- 前回のデバイス プロビジョニング ステータス：Success
- 最新のファブリック プロビジョニング ステータス：Success
- 最新のポリシー プロビジョニング ステータス：Success



Cisco DNA Center 1.1.x では、一部の操作（名前空間）で、各プロビジョニング操作中に同一デバイスに対して一意の ID が作成されます。これにより、デバイス プロビジョニング ステータス Failed が発生します（最初に失敗した後で成功した場合でも）。

Cisco DNA Center 1.2.x 以降でも同じ動作が見られますが、最新のプロビジョニングステータスが個別に表示されます。



次の場合、デバイス プロビジョニング ステータスは常に「Configuring」のままです。

- プロビジョニング ステータス メッセージがサービスでハングする。

  プロビジョニングは非同期動作であるため、影響を受けるサービス（spf-serv、spf-dev、orch-eng、network-programmer）間で、rabbit-mq メッセージを介して通信が行われます。サービスがメッセージを受信できない場合、メッセージは未確認状態でキュー内でハングします。

  たとえば、spf-device manager が spf-service からのメッセージを受信も処理もしない場合は、rabbit-mq コンソールで内容を確認して GUI にログインする必要があります。文字列「spf」でキューをフィルタリングし、未確認メッセージがないか確認します。

- プロビジョニング中に *postgres* サービスが失敗またはクラッシュする。

  各デバイスに対してプロビジョニングタスクが作成されると、システムはステータスが「Configuring」のデータベースエントリを作成します。プロビジョニングが完了すると、spf-serv manager はデータベースエントリを「Failed」または「Success」に更新します。spf-serv manager がステータスを更新しようとしたときにデータベースがクラッシュすると、デバイスのプロビジョニングは「Configuring」でハングします。

  Cisco DNA Center では、常に集約されたデバイス プロビジョニング ステータスが表示されるため、postgres サービスが復元され、プロビジョニング動作が成功した後でも、ステータスは「Configuring」のままです。postgres の再起動回数が最近増加したかどうかを確認する必要があります。

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

この資料の記載内容は2008年10月現在のものです。
この資料に記載された仕様は予告なく変更する場合があります。

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255（フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/