

Cisco Meeting Server(CMS)バージョン3次世代ストリーマおよびアップローダの設定

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[設定](#)

[ネットワーク図](#)

[設定](#)

[確認](#)

[トラブルシューティング](#)

概要

このドキュメントでは、Cisco Meeting Server(CMS)とNext Generation StreamerおよびUploaderの統合を設定およびトラブルシューティングする手順について説明します。次世代ストリーマはCMSバージョン3.0から導入され、Session Initiation Protocol(SIP)ベースです。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- CMS Callbridgeバージョン3.0以降 (Recording/Streamingライセンスあり)。 (1つの録音ライセンスで1つのストリーミングコールが可能)
- Vbrick Distributed Media Engine(DME) (CMSストリーミングサービスからライブストリームをパブリッシュするために使用)
- Vbrick Rev (オプション : ライブストリーミングを内部ネットワークまたはマルチキャストの外部で共有する必要がある場合にのみ必要)
- ネットワークファイルシステム(NFS)ディレクトリが必要で、Windows ServerまたはLinuxでセットアップできます。
- Windowsサーバーの場合は、Windowsでネットワークフ[イル](#)システム(NFS)を展開する手順に従います
- Linuxの場合は、Deploy Network [File](#) system on Linuxの手順に従います

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- 「レコーダー」ライセンスおよび「ストリーマ」ライセンスの両方またはいずれかがある CMS 3.2 以降。 (レコーダーライセンスでは、ストリーミングも可能です)。
- VBrick Distributed Media Engine(DME)3.15.0 RHEL7

- Vbrick Rev(UploaderはvBrick Revサーバで使用されます。手動による録音のインポートは不要)
- Windows Server 2012 R2 (NFS対応)

背景説明

CMSバージョン2.1以降では、標準のReal-Time Messaging Protocol(RTMP)を使用したCMSストリーマによるライブストリーミングがサポートされるようになりました。CMS 3.0では、SIPベースのコンポーネントである次世代ストリーマが導入されました。3.0より前のバージョンでは、Extensible Messaging and Presence Protocol(XMPP)が使用されていました。CMSバージョン3.1以降はRTMPSプロトコルをサポートしているため、CMSストリーマコンポーネントと外部サーバ間の通信は暗号化できます。これにより、CMSストリーマは、RTMP (Youtube、Facebook、Wowzaなど) をサポートするストリーミングプラットフォームと統合できます。現在、CMS Streamerは外部ストリーミングサーバとしてVbrick DMEを使用してテストされており、統合のための推奨プラットフォームです。

VBrick DMEとのライブストリーミング (Webキャスト) 統合により、ユーザはネットワーク内のどこからでも異なるデバイスからライブストリーミングCMS会議を視聴できます。また、VBrick RevがVBrick DMEと共に使用される場合、この機能は内部ネットワークの外部からVBrick Revの許可ユーザごとに拡張されます。また、CMSのアップロードコンポーネントは、Meeting Serverに接続された設定済NFSからビデオコンテンツマネージャVBRICKにMeetingMeetingServerのアップロードのワークフローを手動で録音をインポートする必要はありません。Uploaderコンポーネントを設定して有効にすると、記録がNFSからVbrickにプッシュされます。

ストリーマ

Number of vCPUs	RAM	Number of 720p streams	Number of 1080p streams	Number of audio-only streams
4	4GB	50	37	100
4	8GB	100	75	200
8	8GB	200	150	200

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。ここで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。対象のネットワークが稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

アップローダ

Uploaderコンポーネントは、Recorderコンポーネントと同じサーバ、または別のサーバにインストールできます。レコーダと同じサーバにインストールされている場合は、使用するvCPUを2つ追加します。別のサーバで実行する場合は、レコーダと同じサーバ仕様を使用します。物理コアが4つ以上、RAMが4GB以上の専用VM。

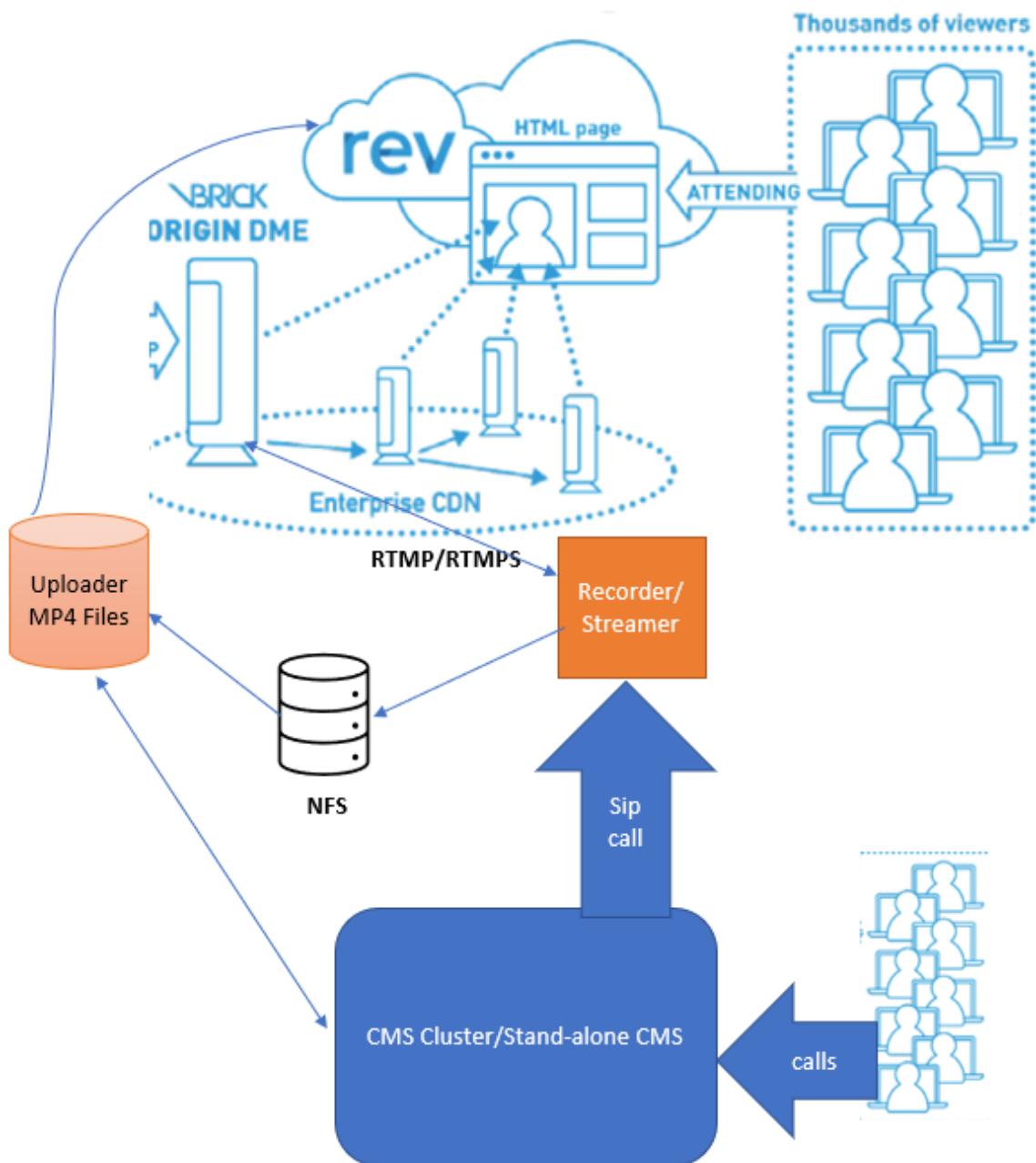
アップロードを実行する会議サーバーには、ネットワークファイル共有(NFS)の読み取りおよび書き込みアクセス許可が必要です。アップローダは、会議をホストするCall Bridgeではなく、別のMeeting Serverで実行する必要があります。

Component	Connecting to	Destination port to open
Call Bridge	NFS (version 3)	2049
Uploader	Web Admin of Call Bridge	443 or port specified in Uploader configuration
Uploader	Vbrick Rev server	443 for video uploads and API access to Vbrick Rev server

設定

ネットワーク図

CMSを使用してストリーマおよびアップローダを導入する場合、次のようなシナリオがサポートされます。複数のストリーミングサーバを備えた単一のコールブリッジ、単一のストリーミングサーバを備えたコールブリッジ クラスタ、複数のストリーミングサーバを備えたコールブリッジ クラスタ。このドキュメントは、単一のストリーマ、アップローダサーバに接続するcallbridgeのクラスタを使用した基本的な展開に基づいており、このシナリオのすべての設定手順が他のシナリオにも適用されます。



上の図に示すように

CMS CallBridgeクラスター

CMSストリーマ/レコーダー

CMSアップローダ

ストリーミング用VBrick DME

ストリーミング用VBrick Rev

設定

ストリーマ

callbridgeはすでに設定されており、コールを受け入れていることを前提としています。

ステップ1：証明書

新しいストリーマコンポーネントでは、https接続をリッスンする必要はありませんが、SIP接続（SIP接続）をリッスンします。ストリーマサーバには、TLS通信用の有効な証明書が必要です。

```
streamer> pki csr tac CN:*.tptac9.com subjectAltName:streamer.tptac9.com .....  
Created key file tac.key and CSR tac.csr CSR file tac.csr ready for download via SFTP
```

ローカル認証局(CA)から署名された証明書を取得します。Secure File Transfer Protocol(SFTP)を使用して生成された証明書をストリーマサーバにアップロードします。オンにすると、証明書が正常にアップロードされたことを確認します。

このドキュメントでは、ストリーマにワイルドカード証明書を使用します。証明書ガイドを参照してください。

https://www.cisco.com/c/dam/en/us/td/docs/conferencing/ciscoMeetingServer/Deployment_Guide/Version-3-1/Certificate-Guidelines-for-all-Deployments-3-1.pdf

```
streamer> pki list User supplied certificates and keys: tac.key tac.cer ROOTCA.cer example.key  
example.csr tac.csr
```

ステップ2:MMP/SSHの設定

- MMPコマンドを使用して、ストリーマのリスニングインターフェイスと、リッスンするSIP TCPポートおよびTLSポートを設定します
ストリーマsip listen <interface> <tcp-port|none> <tls-port|none>

```
streamer > streamer sip listen a 7000
```

TLS sip接続のみを使用する。TCP sip接続を「none」として設定します。コマンドは次のとおりです

```
streamer > streamer sip listen a none 7001
```

- ストリーマサーバに証明書を適用する
ストリーマ> streamer sip certs tac.key tac.cer

- ストリーミングの品質を選択します

streamer > streamer sip resolution 720p

- ストリーマの有効化
streamer> streamer enable

- オプションで、TLSが設定されている場合、ストリーマでSIPのTLS検証を実行できます
streamer > tls sip trust ROOTCA.cer

注:TLS接続を保護するには、TLS検証を有効にすることを推奨します。
ストリーマ> tls sip verify enable

- 上記の設定情報が正しいことを確認します

```
streamer> streamer Enabled : false SIP interfaces : tcp a:7000, tls a:7001 SIP key file :  
tac.key SIP certificate file : tac.cer SIP CA Bundle file : none SIP Resolution : 720p SIP  
traffic trace : Disabled Call Limit : none
```

- 次のコマンドを使用して、ストリーマを有効にします。「**streamer enable**」すべてのメッセージに次のように「SUCCESS」と表示される必要があります

```
streamer> streamer enable SUCCESS: Key and certificate pair match SUCCESS: Streamer enabled
```

ステップ3:APIの設定

この設定は、CallBridgeをホストするCMSで実行されます。次に、CMSのWebadminインターフェイスのAPIを示します。Postman、PosterなどのRESTクライアントも使用できます。

新しいSIPストリーマを有効にすると、sipStreamUriの/callProfilesの下のCall Bridgeで設定および使用できます。

CMSサーバでAPIを使用する。[Webadmin] > [Configuration] > [API]に移動します。

- コールプロファイルの作成

« start < prev 1 - 1 (of 1) next »

show all ▼

Create new

Table view

XML view

object id

- sipStreamUri=demo@streamer.com & StreamingMode=自動/手動の設定

The user part of the configured "sipStreamUri" (i.e. the part before '@' symbol) has no significant meaning, and for the new SIP streamer component, although required, it can usually be anything, e.g. "streamer@streamer.com".The important part of the URI is the "domain" part.

[« return to object list](#)

/api/v1/callProfiles/1c390ade-d4c2-4cfb-bc3a-251dc6d8b367

Related objects: [/api/v1/callProfiles](#)

[Table view](#) [XML view](#)

Object configuration

streamingMode manual
sipStreamerUri demo@streamer.com

Write this object to "/api/v1/system/profiles"

/api/v1/callProfiles/1c390ade-d4c2-4cfb-bc3a-251dc6d8b367

participantLimit	<input type="checkbox"/>	<input type="text"/>
locked	<input type="checkbox"/>	<unset> v
recordingMode	<input type="checkbox"/>	<unset> v
streamingMode	<input type="checkbox"/>	manual v - present
passcodeMode	<input type="checkbox"/>	<unset> v
passcodeTimeout	<input type="checkbox"/>	<input type="text"/>
gatewayAudioCallOptimization	<input type="checkbox"/>	<unset> v
lyncConferenceMode	<input type="checkbox"/>	<unset> v
lockMode	<input type="checkbox"/>	<unset> v
sipRecorderUri	<input type="checkbox"/>	<input type="text"/>
sipStreamerUri	<input type="checkbox"/>	demo@streamer.com - present
		<input type="button" value="Modify"/>

sipStreamuri=demo@streamer.com & StreamingMode=自動/手動の設定

- 上に作成したストリーマcallProfileを/system/profilesに追加します。これはグローバル設定であり、設定された「sipStreamerUri」がストリーマの動作に使用されます。

[« return to object list](#)

/api/v1/system/profiles

Related objects: </api/v1/system/profiles/effectiveWebBridgeProfile>

[Table view](#) [XML view](#)

Object configuration	
callLegProfile	add951de-3ded-4619-a428-d779e6b0a323
callProfile	1c390ade-d4c2-4cfb-bc3a-251dc6d8b367
dtmfProfile	5a7e4d42-adc7-4ad5-8bc3-e3b998e2d648

/api/v1/system/profiles

callLegProfile	<input type="checkbox"/>	<input type="text" value="add951de-3ded-4619-a428-d779e6b0a323"/>	<input type="button" value="Choose"/> - present
callProfile	<input type="checkbox"/>	<input type="text" value="1c390ade-d4c2-4cfb-bc3a-251dc6d8b367"/>	<input type="button" value="Choose"/> - present
dtmfProfile	<input type="checkbox"/>	<input type="text" value="5a7e4d42-adc7-4ad5-8bc3-e3b998e2d648"/>	<input type="button" value="Choose"/> - present
userProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>
ivrBrandingProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>
callBrandingProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>
compatibilityProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>
dialInSecurityProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>
webBridgeProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>

システム/プロフィールへのcallProfileの追加

- ストリーミングに使用するスペースにVBrick 'streamURL'を追加します。'Stream'という名前のスペースは、CMS Webインターフェイスを使用して作成しました

Space configuration

Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout	
<input type="checkbox"/> Stream	1004			98765		not set	[edit]
<input type="checkbox"/> TAC2	1005			654		not set	[edit]
<input type="checkbox"/> Telepresence	3005					not set	[edit]
<input type="checkbox"/> Telepresence	3001					not set	[edit]

ストリーミングに使用する領域

- スペースを変更して「StreamURL」を追加します。'streamURL'の形式は次のとおりです。
`rtmp://<VBrickBroadcastUsername>:<VBrickBroadcastPassword>@<VBrick IPまたはFQDN>/live/NameofStream`例では、これを「`rtmp://broadcast:broadcast@10.106.81.40/live/SpaceStream77`」として設定しています

Object configuration	
name	Stream
autoGenerated	false
uri	1004
callId	98765
streamUri	rtmp://broadcast:broadcast@10.106.81.40/live/SpaceStream77
secret	VWzUQlu5cuqUG2j8.p9R_g

/api/v1/coSpaces/ca2847f3-02c7-438b-93e5-09434ec9de42

userProvisionedCoSpace	<input type="checkbox"/>		GUID (none available)
name	<input type="checkbox"/>	Stream	- present
uri	<input type="checkbox"/>	1004	(URI user part) - present
secondaryUri	<input type="checkbox"/>		(URI user part)
callId	<input type="checkbox"/>	98765	- present
cd/Tag	<input type="checkbox"/>		
passcode	<input type="checkbox"/>		
defaultLayout	<input type="checkbox"/>	<unset>	
tenant	<input type="checkbox"/>		Choose
callLegProfile	<input type="checkbox"/>		Choose
callProfile	<input type="checkbox"/>		Choose
calBrandingProfile	<input type="checkbox"/>		Choose
dialInSecurityProfile	<input type="checkbox"/>		Choose
requireCallId	<input type="checkbox"/>	<unset>	
secret	<input type="checkbox"/>	VWzUQlu5cuqUG2j8.p9R_g	- present
regenerateSecret	<input type="checkbox"/>	<unset>	
nonMemberAccess	<input type="checkbox"/>	<unset>	
ownerJid	<input type="checkbox"/>		
streamUri	<input type="checkbox"/>	rtmp://broadcast broadcast@10.106.81.40/live/SpaceStream77	(URL) - present
ownerAdGuid	<input type="checkbox"/>		GUID (none available)
meetingScheduler	<input type="checkbox"/>		
panePlacementHighestImportance	<input type="checkbox"/>		
panePlacementSelfPaneMode	<input type="checkbox"/>	<unset>	
Modify			

ステップ 4 : 「アウトバウンドルール」の作成

outboundDialPlanルールにマッピングするカスタムURIを設定します (ドメインは「streamer.com」など任意のURIにすることができます)。streamerUriで使用されるドメインをルーティングするようにoutboundDialPlanルールを設定してください。

Outbound calls

Filter	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	Call Bridge Scope		
	<input type="checkbox"/>	streamer.com	10.106.81.58:7000		<use local contact domain>	Standard SIP	Continue	20	Unencrypted	no	<all>	[edit]
	<input type="checkbox"/>	streamer.com	10.106.81.58:7001		<use local contact domain>	Standard SIP	Stop	20	Encrypted	no	<all>	[edit]
	<input type="checkbox"/>	recorder.com	10.106.81.58:6060		<use local contact domain>	Standard SIP	Continue	10	Unencrypted	no	<all>	[edit]
	<input type="checkbox"/>	recorder.com	10.106.81.58:6061		<use local contact domain>	Standard SIP	Stop	10	Encrypted	no	<all>	[edit]
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Standard SIP	Stop	0	Auto				Add New Reset

アウトバウンドルールの作成

SIPSIP(5060,5061)DialPlansip proxy to use

```
streamer> streamer Enabled : true SIP interfaces : tcp a:7000, tls a:7001 SIP key file : tac.key
SIP certificate file : tac.cer SIP CA Bundle file : none SIP Resolution : 720p SIP traffic trace
: Disabled Call Limit : none
```

アップローダ

- アップローダがモニタする録音を保存するNFSおよびディレクトリを指定します
streamer> uploader nfs 192.168.15.38 : 録音

- アップロード者が記録情報を照会する会議サーバを指定します
streamer> uploader cms host join.mextp.local

- Call Bridgeを実行しているMeeting ServerのWeb管理ポートを指定します

```
streamer > uploader cms port 445
```

- Call Bridgeを実行しているMeeting ServerでAPIアクセス権を持つユーザを指定します

```
streamer > uploader cms user apiadmin
```

```
streamer > uploader cms password
```

パスワードを入力してください :

- CMSからの証明書バンドルをMeeting Server信頼ストアに追加します

Call Bridgeを実行しているMeeting Serverで、ルートCAの証明書とすべての中間証明書のコピーをチェーンに保持する証明書バンドル(cert-bundle)を作成します。

```
streamer > uploader cms trust ROOTCA.cer
```

- Vbrickホストと、アップローダが接続するポートを設定します

```
streamer> uploader rev host ciscotac.rev-na.demo.vbrick.com
```

```
streamer> uploader rev port 443
```

注 : 特に指定がない限り、ポートはデフォルトで443に設定されます。

- ビデオ録画をアップロードするAPI権限を持つVbrick Revユーザを追加する

```
streamer> uploader rev user tacuser
```

```
streamer> uploader rev password
```

パスワードを入力してください :

- 証明書バンドルをVbrick Rev信頼ストアに追加します

ルートCA証明書とすべての中間証明書のコピーを保持する証明書バンドル(cert-bundle)をVbrick Revサーバのチェーンに作成します

```
streamer> uploader rev trust vbrickbundle.cer
```

- アップローダの設定をチェックし、アップローダを有効にします

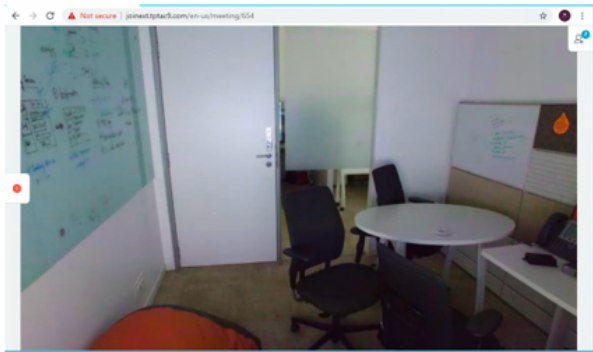
```
streamer> uploader Enabled : false NFS hostname : 192.168.15.38 NFS directory : Recording CMS
host : join.mextp.local CMS port : 445 CMS user : apiadmin CMS trust bundle : ROOTCA.cer Vbrick
Rev hostname : ciscotac.rev-na.demo.vbrick.com Vbrick Rev port : 443 Vbrick Rev username :
tacuser Vbrick Rev trust bundle : brick.cer View access : Public cospace_member_access : edit
recording_owned_by_cospace_owner : false fallback_owner : admin comments_enabled : true
ratings_enabled : true downloads_enabled : true active_upon_upload : true delete_after_upload :
false
```

設定が正しい場合は、「**uploader enable**」コマンドを使用してUploaderコンポーネントを有効にします。次に示すように、すべてのメッセージに「SUCCESS」と表示される必要があります。

```
streamer> uploader enable SUCCESS: uploader enabled
```

確認

ストリーマ



Conference: Stream (3 active calls)	
<input type="checkbox"/>	SIP 9596@10.127.241.219 [less] (call 77, incoming, unencrypted) call duration 2 minutes, 28 seconds incoming media AAC (64.0 Kb/s, -6.0dB gain), H.264, 768 x 448 30.0fps, 321 Kb/s outgoing media OPUS, H.264, 640 x 360 4.9fps, 4.91 Kb/s additional protocols unencrypted Active Control, unencrypted H.224 remote address 9596@10.127.241.219 SIP call ID 1cd1b000-10001-291f-93f492a@10.127.241.219
<input type="checkbox"/>	SIP demo@streamer.com [less] (call 81, outgoing, unencrypted) call duration 9 seconds incoming media no audio outgoing media AAC, H.264, 1280 x 720 29.9fps, 1.89 Mb/s remote address demo@streamer.com SIP call ID 18cc4f61-1af3-41b0-bdff-fe5378e2265b
<input type="checkbox"/>	CMA Guest_390 [less] (call 80, incoming, encrypted - AES-128) call duration 27 seconds incoming media no audio outgoing media OPUS, H.264, 768 x 448 30.0fps, 321 Kb/s remote address guest2537753157

Connected Multi-Protocol Streams							Tue, 11 Aug 2020 06:34:16
Stream Type	IP Address	Port	Packets/Segments Sent	Packets/Segments Lost	Time Connected	Connected To	
In RTMP Active	10.106.81.58	1935	14840	0	2 min 23 sec	SpaceStream77	
rtsp://10.106.81.40:5544/SpaceStream77 rtsp://10.106.81.40:5544/SpaceStream77 rtmp://10.106.81.40:1935/live/SpaceStream77							

ストリーミング動作とSIPストリーミングコールが接続

アップローダ

uploaderのsyslog追跡で、成功したイベントのログを確認できます。

```
Jun 17 22:24:41.867 user.info cms-02 Uploader[1]: scanning directory:
/mnt/recordings/forwardedCalls Jun 17 22:24:41.867 user.info cms-02 Uploader[1]: scanning
directory: /mnt/recordings/spaces Jun 17 22:24:41.869 user.info cms-02 Uploader[1]: checking the
status of /mnt/recordings/spaces/8a7076e2-6db6-47e9-98ee-3bd063e32559/20210618032309+0000_vid-
id=c4605aaf-dc49-4cd7-9174-c46185ba1983@vbrick.mp4 Jun 17 22:24:41.870 user.info cms-02
Uploader[1]: Getting from: https://ciscotac.rev-na.demo.vbrick.com:443/api/v1/videos/c4605aaf-
dc49-4cd7-9174-c46185ba1983/status Jun 17 22:24:42.035 user.info cms-02 Uploader[1]: Received
vbrick response status code: 200 Jun 17 22:24:42.035 user.info cms-02 Uploader[1]: vbrick
response: main.vbrickStatusResp{Status:"Ready"} Jun 17 22:24:42.035 user.info cms-02
Uploader[1]: file 20210618032309+0000_vid-id=c4605aaf-dc49-4cd7-9174-c46185ba1983@vbrick.mp4 vid
c4605aaf-dc49-4cd7-9174-c46185ba1983 status Ready Jun 17 22:24:42.035 user.info cms-02
Uploader[1]: Getting from: https://ciscotac.rev-na.demo.vbrick.com:443/api/v1/videos/c4605aaf-
dc49-4cd7-9174-c46185ba1983/playback-url Jun 17 22:24:42.200 user.info cms-02 Uploader[1]:
Received vbrick response 200
```

トラブルシューティング

ストリーマ

1.ライセンスなし

ストリーマには、callbridgeコンポーネントを備えたサーバ上で「レコーダ」ライセンスが必要でした。存在しない場合、またはライセンスが不足している場合は、次に示すエラーイベントログに表示されます。

2020-08-09	04:00:18.946	Info	API "TAC2" Space GUID: d4f2fa4c-4730-4dcb-9bcb-1d65c9e59016 <--> Call Correlator GUID: fceebc78-ba2f-4f6d-8c29-cd711cc6de09 <--> Internal GUI
2020-08-09	04:00:18.946	Info	conference d1be8391-ed8c-4406-994c-2e07b46fbf4c: lock state has changed to unlocked
2020-08-09	04:00:18.946	Info	API call leg d95ba532-16c3-4afe-bf82-5514c9219efd in call d1be8391-ed8c-4406-994c-2e07b46fbf4c (API call 5d9067f5-44fd-4a63-8a36-af2d43a09cad)
2020-08-09	04:00:18.946	Info	unable to start recording (space 'TAC2') -- no license
2020-08-09	04:00:18.947	Info	conference d1be8391-ed8c-4406-994c-2e07b46fbf4c has control/media GUID: 6d365821-ddfa-49d7-bf4c-323089c2c3e5
2020-08-09	04:00:18.947	Info	conference d1be8391-ed8c-4406-994c-2e07b46fbf4c named "TAC2"
2020-08-09	04:00:18.947	Info	call 2: configured - API call leg d95ba532-16c3-4afe-bf82-5514c9219efd with SIP call ID "58dda880-10001-21-93f492a@10.127.241.219"
2020-08-09	04:00:19.213	Info	call 2: compensating for far end not matching payload types

必要なライセンスを追加してください。ライセンスのステータスは、コマンド「license」を使用してCLIで確認できます

```
cms1> license Feature: callbridge status: Activated expiry: 2023-Apr-28 (690 days remain)
Feature: turn status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: webbridge status:
Activated expiry: 2023-Apr-28 (690 days remain) Feature: customizations status: Activated
expiry: 2023-Apr-28 (690 days remain) Feature: local_license_mode status: Activated expiry:
2023-Apr-28 (690 days remain) Feature: recording status: Activated expiry: 2023-Apr-28 (690 days
remain) Feature: personal status: Activated expiry: 2023-Apr-28 (690 days remain) Feature:
shared status: Activated expiry: 2023-Apr-28 (690 days remain)
```

2. TLSポート

- ・ TLSポートが設定されていて、証明書が適用されていない場合。TLSを使用するためのストリーマの証明書の設定
- ・ 証明書が使用できない場合。TCPポートだけを設定する

```
streamer> streamer sip listen a 7000 7001 streamer> streamer enable FAILURE: TLS port set but no
certificates configured FAILURE: Streamer configuration not complete
```

これで、TLSポートを削除するか、SIP TLS信頼およびストリーマ証明書を追加する、2つのオプションが用意されました

TLSポートを有効にすることを推奨します。

3. RTMPストリームが正しく設定されていない

ログにエラーが表示されます

```
daemon.info streamer streamer-sip[2280]: 144500.368 : INFO : call 3: retrieved stream URL from
RTCP: "rtmp://broadcast:broadcast@10.106.81.40/test" daemon.info streamer streamer-sip[2280]:
144500.368 : INFO : call 3: parsing rtmp://broadcast:broadcast@10.106.81.40/test daemon.info
streamer streamer-sip[2280]: 144500.368 : INFO : call 3: RTMP stream="test" daemon.info streamer
streamer-sip[2280]: 144500.368 : INFO : call 3: RTMP server="rtmp://10.106.81.40:1935/test"
daemon.info streamer streamer-sip[2280]: 144500.370 : INFO : call 3: Connected to RTMP server
daemon.info streamer streamer-sip[2280]: 144500.370 : INFO : call 3: C2 pending - len 1536
daemon.info streamer streamer-sip[2280]: 144500.370 : : call 3: snd: create new chunk stream 2
daemon.info streamer streamer-sip[2280]: 144500.370 : : call 3: snd: create new chunk stream 3
daemon.info streamer streamer-sip[2280]: 144500.370 : INFO : call 3: RTMP sent chunk size of
4096 and connect message daemon.info streamer streamer-sip[2280]: 144500.410 : ERROR : call 3:
connection : far end closed connection 5
```

ストリーマ設定の手順を確認し、RTMP URLを「rtmp://<VBrickBroadcastUsername>:<VBrickBroadcastPassword>@<VBrick IPまたはFQDN>/live/NameofStream」の形式で正しく設定してください。

4. コールルーティングに関する問題

CMSストリーマはSIPベースのクライアントであるため、前述したように、ルーティングを設定する必要があります。これにより、コールが失敗するシナリオが発生する可能性があります。この例では、CMS Callbridgeが発信コールを送信したが、次の「トランザクションタイムアウト - 暫定応答なし送信 INVITE」エラーで失敗しました

```
2021-06-28 17:37:02.412 Info user 'guest300535034' starting streaming (space 'test')
2021-06-28 17:37:02.413 Info API call leg bc0917df-589c-4628-887d-79481d322fed in call
63f0b174-831e-4a12-b4ee-27186d4162af (API call 00286960-9af9-4d5d-9ca7-20dd40425292)
2021-06-28 17:37:02.413 Info call 44: outgoing SIP call to "demo@streamer.com" from space
"test"
2021-06-28 17:37:02.413 Info call 44: configured - API call leg bc0917df-589c-4628-887d-
79481d322fed with SIP call ID "7d37a80e-7996-4e8d-aa87-77c9d4729cec"
2021-06-28 17:37:04.482 Info call 42: receiver report 1 interval for rx video 0 = 6113ms
(period 6108ms) 00000000
2021-06-28 17:37:22.074 Info call 44: falling back to unencrypted control connection...
2021-06-28 17:37:54.075 Info call 44: ending; local SIP teardown with reason 7
(transaction timeout - no provisional responses sending INVITE) - not connected after 0:52
2021-06-28 17:37:54.075 Info call 44: destroying API call leg bc0917df-589c-4628-887d-
79481d322fed
2021-06-28 17:37:54.076 Info streaming call leg for space 'test' disconnected with reason
7 (transaction timeout - no provisional responses sending INVITE)
CMS Callbridgeサーバの[発信コール(Outbound Calls)]設定を確認し、送信先のロケーションと正しく設定されているかどうかを確認します。また、コール
プロファイルが正しいストリーマURIで設定されていて、同じプロファイルがCospaceに関連付けられているかどうかを確認します。
```

アップローダ

1.ブリックの詳細が正しくありません。

アップロードログにエラーが表示されます

```
Jun 27 11:29:27.864 user.info streamer Uploader[1]: Received vbrick response 500 Jun 27
11:29:27.864 user.info streamer Uploader[1]: posting to:
https://sales.vbrick.com:443/api/v1/user/login Jun 27 11:29:47.870 user.info streamer
Uploader[1]: Received vbrick response 500 Jun 27 11:29:47.870 user.err streamer Uploader[1]:
Failed to initialise Vbrick Client Jun 27 11:29:47.870 user.err streamer Uploader[1]: vbrick
returned status code: 500
```

vbrickサーバに正しいクレデンシャルとポートが設定されていることを確認します。 また、アップローダがCMS callbridge webadminポートに到達できることを確認します。