# 音声サービスコンテナのよくある質問（FAQ）

* 2021/03/11

コンテナーでSpeechサービスを使用する場合は、サポートにエスカレーションする前に、このよくある質問のコレクションに依存してください。この記事では、一般的なものから技術的なものまで、さまざまな程度の質問を取り上げます。回答を展開するには、質問をクリックしてください。

## 一般的な質問

### 音声コンテナはどのように機能し、どのように設定しますか？

本番クラスターをセットアップするときは、考慮すべきことがいくつかあります。まず、同じマシン上に単一の言語、複数のコンテナーをセットアップすることは大きな問題ではありません。問題が発生している場合は、ハードウェア関連の問題である可能性があります。そのため、最初にリソースを調べます。CPUとメモリの仕様。

ja-JPコンテナと最新モデルについて少し考えてみてください。音響モデルはCPUに関して最も要求の厳しい部分ですが、言語モデルは最も多くのメモリを要求します。使用のベンチマークを行ったところ、音声がリアルタイムで（マイクからのように）流入している場合、単一の音声からテキストへのリクエストを処理するのに約0.6CPUコアが必要です。（ファイルからのように）リアルタイムよりも速くオーディオを供給している場合、その使用量は2倍になる可能性があります（1.2xコア）。一方、以下のメモリは音声をデコードするための動作メモリです。ファイルキャッシュに存在する言語モデルの実際のフルサイズは考慮されません。それja-JPは追加の2GBです。の場合en-US、それ以上になる可能性があります（6〜7GB）。

メモリが不足しているマシンを使用していて、そのマシンに複数の言語をデプロイしようとしている場合、ファイルキャッシュがいっぱいになり、OSがモデルのページインとページアウトを強制される可能性があります。実行中の文字起こしの場合、これは悲惨な結果になる可能性があり、速度低下やその他のパフォーマンスへの影響につながる可能性があります。

さらに、[高度なベクトル拡張（AVX2）](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-howto#advanced-vector-extension-support)命令セットを備えたマシンの実行可能ファイルを事前にパッケージ化します。AVX512命令セットを備えたマシンでは、そのターゲットのコード生成が必要になり、10言語で10個のコンテナーを起動すると、一時的にCPUが使い果たされる可能性があります。次のようなメッセージがDockerログに表示されます。

コンソールコピー

2020-01-16 16:46:54.981118943

[W:onnxruntime:Default, tvm\_utils.cc:276 LoadTVMPackedFuncFromCache]

Cannot find Scan4\_llvm\_\_mcpu\_skylake\_avx512 in cache, using JIT...

変数を使用して、単一のコンテナー内に必要なデコーダーの数を設定できDECODER MAX\_COUNTます。したがって、基本的には、SKU（CPU /メモリ）から始める必要があり、それを最大限に活用する方法を提案できます。優れた出発点は、推奨されるホストマシンのリソース仕様を参照することです。

### オンプレミスのSpeech-to-Textコンテナの容量計画とコスト見積もりを手伝ってもらえますか？

バッチ処理モードでのコンテナ容量の場合、各デコーダーは2つのCPUコアを使用して、1回の認識で2〜3倍をリアルタイムで処理できます。コンテナインスタンスごとに3つ以上の同時認識を維持することはお勧めしませんが、信頼性/可用性の理由から、ロードバランサの背後でコンテナのインスタンスをさらに実行することをお勧めします。

ただし、各コンテナインスタンスをより多くのデコーダーで実行することもできます。たとえば、8コアマシンのコンテナインスタンスごとに7つのデコーダーを（それぞれ2倍以上で）セットアップして、15倍のスループットを実現できる場合があります。DECODER\_MAX\_COUNT注意すべきパラメータがあります。極端な場合、スループットが大幅に向上し、信頼性と遅延の問題が発生します。マイクの場合、リアルタイムの1倍になります。全体的な使用量は、1回の認識で約1コアである必要があります。

バッチ処理モードで1日1K時間処理するシナリオの場合、極端な場合、3つのVMが24時間以内に処理できますが、保証はされません。スパイク日、フェイルオーバー、更新を処理し、最小限のバックアップ/ BCPを提供するには、クラスターごとに3台ではなく、4〜5台のマシンを使用し、クラスターが2つ以上あることをお勧めします。

ハードウェアについては、標準のAzure VMDS13\_v2を参照として使用します（各コアは2.6 GHz以上で、AVX2命令セットが有効になっている必要があります）。

| **表1** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **インスタンス** | **vCPU** | **羊** | **一時保管** | **AHBで従量課金制** | **AHBによる1年間の準備金（％節約）** | **AHBで3年間予約（％節約）** |
| DS13 v2 | 8 | 56 GiB | 112 GiB | $ 0.598 /時間 | $ 0.3528 /時間（〜41％） | $ 0.2333 /時間（〜61％） |

設計リファレンス（1 K時間/日のオーディオバッチ処理を処理する5つのVMの2つのクラスター）に基づくと、1年間のハードウェアコストは次のようになります。

2（クラスター）\* 5（クラスターあたりのVM）\* $ 0.3528 /時間\* 365（日）\* 24（時間）= $ 31K /年

物理マシンにマッピングする場合、一般的な見積もりは1 vCPU = 1物理CPUコアです。実際には、1vCPUはシングルコアよりも強力です。

オンプレミスの場合、次のすべての追加要素が関係します。

* 物理CPUの種類と、その上のコアの数
* 同じボックス/マシンで一緒に実行されているCPUの数
* VMの設定方法
* ハイパースレッディング/マルチスレッドの使用方法
* メモリの共有方法
* OS等

通常、Azure環境ほど適切に調整されていません。他のオーバーヘッドを考慮すると、安全な見積もりは10個の物理CPUコア= 8個のAzurevCPUです。一般的なCPUには8つのコアしかありませんが。オンプレミス展開では、AzureVMを使用するよりもコストが高くなります。また、減価償却率も考慮してください。

サービス費用はオンラインサービスと同じです。音声認識の場合は1時間あたり1ドルです。音声サービスの費用は次のとおりです。

$ 1 \* 1000 \* 365 = $ 365K

マイクロソフトに支払われる保守費用は、サービスレベルとサービスの内容によって異なります。基本レベルの月額29.99ドルから、オンサイトサービスが含まれる場合は数十万ドルまでさまざまです。おおよその数値は、サービス/保守のために1時間あたり300ドルです。人件費は含まれていません。その他のインフラストラクチャコスト（ストレージ、ネットワーク、ロードバランサーなど）は含まれていません。

### 音声文字変換に句読点がないのはなぜですか？

speech\_recognition\_language=<YOUR\_LANGUAGE>彼らはカーボンクライアントを使用している場合は、明示的に要求して設定する必要があります。

例えば：

Pythonコピー

if not recognize\_once(

speechsdk.SpeechRecognizer(

speech\_config=speechsdk.SpeechConfig(

endpoint=template.format("interactive"),

speech\_recognition\_language="ja-JP"),

audio\_config=audio\_config)):

print("Failed interactive endpoint")

exit(1)

出力は次のとおりです。

cmdコピー

RECOGNIZED: SpeechRecognitionResult(

result\_id=2111117c8700404a84f521b7b805c4e7,

text="まだ早いまだ早いは猫である名前はまだないどこで生まれたかとんと見当を検討をなつかぬ。

何でも薄暗いじめじめした所でながら泣いていた事だけは記憶している。

まだは今ここで初めて人間と言うものを見た。

しかも後で聞くと、それは書生という人間中で一番同額同額。",

reason=ResultReason.RecognizedSpeech)

### カスタム音響モデルと言語モデルをSpeechコンテナで使用できますか？

現在、カスタム言語モデルまたはカスタム音響モデルのいずれか1つのモデルIDのみを渡すことができます。

音響モデルと言語モデルの両方を同時にサポートしないという決定がなされました。これは、APIの中断を減らすために統一された識別子が作成されるまで有効です。したがって、残念ながら、これは現在サポートされていません。

### カスタムの音声テキストコンテナからこれらのエラーを説明できますか？

**エラー1：**

cmdコピー

Failed to fetch manifest: Status: 400 Bad Request Body:

{

"code": "InvalidModel",

"message": "The specified model is not supported for endpoint manifests."

}

最新のカスタムモデルでトレーニングしている場合、現在サポートされていません。古いバージョンでトレーニングする場合は、使用できるはずです。現在も最新バージョンのサポートに取り組んでいます。

基本的に、カスタムコンテナは、HalideまたはONNXベースの音響モデル（カスタムトレーニングポータルのデフォルト）をサポートしていません。これは、カスタムモデルが暗号化されていないためであり、ONNXモデルを公開したくないためです。言語モデルは問題ありません。お客様は、カスタムトレーニング用に古い非ONNXモデルを明示的に選択する必要があります。精度は影響を受けません。モデルサイズはもっと大きくなる場合があります（100MB）。

サポートモデル> 20190220（v4.5 Unified）

**エラー2：**

cmdコピー

HTTPAPI result code = HTTPAPI\_OK.

HTTP status code = 400.

Reason: Synthesis failed.

StatusCode: InvalidArgument,

Details: Voice does not match.

リクエストでは、大文字と小文字が区別される正しい音声名を指定する必要があります。完全なサービス名のマッピングを参照してください。

**エラー3：**

JSONコピー

{

"code": "InvalidProductId",

"message": "The subscription SKU \"CognitiveServices.S0\" is not supported in this service instance."

}

コグニティブサービスリソースではなく、スピーチリソースを作成する必要があります。

### RESTまたはWSのどのAPIプロトコルがサポートされていますか？

音声からテキストへのコンテナおよびカスタムの音声からテキストへのコンテナについては、現在、WebSocketベースのプロトコルのみをサポートしています。SDKはWSでの呼び出しのみをサポートし、RESTはサポートしません。RESTサポートを追加する計画がありますが、現時点ではETAは追加しません。常に公式ドキュメントを参照してください。[クエリ予測エンドポイントを](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-howto#query-the-containers-prediction-endpoint)参照してください。

### CentOSはSpeechコンテナでサポートされていますか？

CentOS7はPythonSDKでまだサポートされていません。また、Ubuntu19.04もサポートされていません。

Python Speech SDKパッケージは、次のオペレーティングシステムで利用できます。

* **Windows** -x64およびx86
* **Mac** -macOSXバージョン10.12以降
* **Linux** -Ubuntu 16.04、Ubuntu 18.04、x64上のDebian 9

環境設定の詳細については、[Pythonプラットフォームの設定を](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/quickstarts/setup-platform?pivots=programming-language-python)参照してください。今のところ、Ubuntu18.04が推奨バージョンです。

### LUIS予測エンドポイントを呼び出そうとするとエラーが発生するのはなぜですか？

IoT EdgeデプロイメントでLUISコンテナーを使用していて、別のコンテナーからLUIS予測エンドポイントを呼び出そうとしています。LUISコンテナはポート5001でリッスンしており、使用しているURLは次のとおりです。

C＃コピー

var luisEndpoint =

$"ws://192.168.1.91:5001/luis/prediction/v3.0/apps/{luisAppId}/slots/production/predict";

var config = SpeechConfig.FromEndpoint(new Uri(luisEndpoint));

私が得ているエラーは次のとおりです。

cmdコピー

WebSocket Upgrade failed with HTTP status code: 404 SessionId: 3cfe2509ef4e49919e594abf639ccfeb

LUISコンテナログにリクエストが表示され、メッセージには次のように表示されます。

cmdコピー

The request path /luis//predict" does not match a supported file type.

これは何を意味するのでしょうか？何が足りないのですか？[ここ](https://github.com/Azure-Samples/cognitive-services-speech-sdk)から、SpeechSDKの例に従い[ました](https://github.com/Azure-Samples/cognitive-services-speech-sdk)。シナリオでは、トレーニングしたLUISアプリに基づいて、PCマイクから直接オーディオを検出し、意図を判断しようとしています。私がリンクした例はまさにそれをします。また、LUISクラウドベースのサービスとうまく連携します。Speech SDKを使用すると、speech-to-text APIを明示的に呼び出してから、LUISを2回呼び出す必要がなくなるようです。

ですから、私がやろうとしているのは、クラウドでLUISを使用するシナリオからLUISコンテナーを使用するシナリオに切り替えることだけです。Speech SDKが一方に機能するかどうかは想像できませんが、もう一方には機能しません。

スピーチSDKはLUISコンテナに対して使用しないでください。LUISコンテナを使用するには、LUISSDKまたはLUISRESTAPIを使用する必要があります。音声SDKは、音声コンテナに対して使用する必要があります。

クラウドはコンテナとは異なります。クラウドは、複数の集約されたコンテナー（マイクロサービスと呼ばれることもあります）で構成できます。つまり、LUISコンテナがあり、次に音声コンテナがあります-2つの別々のコンテナ。音声コンテナは音声のみを実行します。LUISコンテナはLUISのみを実行します。クラウドでは、両方のコンテナーがデプロイされていることがわかっており、リモートクライアントがクラウドに移動して音声を再生し、戻ってから再びクラウドに移動してLUISを実行するのはパフォーマンスが悪いため、次のことができる機能を提供します。クライアントはSpeechに移動し、クラウドにとどまり、LUISに移動してから、クライアントに戻ります。したがって、このシナリオでも、Speech SDKは音声付きのSpeechクラウドコンテナに移動し、Speechクラウドコンテナはテキスト付きのLUISクラウドコンテナと通信します。LUISコンテナには、オーディオを受け入れるという概念がありません（LUISコンテナがストリーミングオーディオを受け入れることは意味がありません。LUISはテキストベースのサービスです）。オンプレミスでは、お客様が両方のコンテナを展開したかどうかはわかりません。お客様の敷地内のコンテナ間でオーケストレーションを行うとは想定していません。両方のコンテナがオンプレミスで展開されている場合は、クライアントにとってよりローカルであるため、最初にSRに移動し、クライアントに戻ってから、顧客にそのテキストを取得してLUISに移動させるのは負担ではありません。

### macOS、Speechコンテナ、Python SDKでエラーが発生するのはなぜですか？

転記する.wavファイルを送信すると、結果は次のように返されます。

cmdコピー

recognition is running....

Speech Recognition canceled: CancellationReason.Error

Error details: Timeout: no recognition result received.

When creating a websocket connection from the browser a test, we get:

wb = new WebSocket("ws://localhost:5000/speech/recognition/dictation/cognitiveservices/v1")

WebSocket

{

url: "ws://localhost:5000/speech/recognition/dictation/cognitiveservices/v1",

readyState: 0,

bufferedAmount: 0,

onopen: null,

onerror: null,

...

}

WebSocketが正しく設定されていることはわかっています。

その場合は、[このGitHubの問題を](https://github.com/Azure-Samples/cognitive-services-speech-sdk/issues/310)参照[して](https://github.com/Azure-Samples/cognitive-services-speech-sdk/issues/310)ください。[ここ](https://github.com/Azure-Samples/cognitive-services-speech-sdk/issues/310#issuecomment-527542722)で[提案されている](https://github.com/Azure-Samples/cognitive-services-speech-sdk/issues/310#issuecomment-527542722)回避[策があります](https://github.com/Azure-Samples/cognitive-services-speech-sdk/issues/310#issuecomment-527542722)。

Carbonはこれをバージョン1.8で修正しました。

### Speechコンテナのエンドポイントの違いは何ですか？

テストする関数や、SDKおよびREST APIをテストする方法など、次のテスト指標の入力を手伝ってもらえますか？特に、既存のドキュメント/サンプルでは見られなかった「インタラクティブ」と「会話」の違い。

| **表2** | | | |
| --- | --- | --- | --- |
| **終点** | **機能テスト** | **SDK** | **REST API** |
| /speech/synthesize/cognitiveservices/v1 | テキストの合成（テキスト読み上げ） |  | はい |
| /speech/recognition/dictation/cognitiveservices/v1 | コグニティブサービスのオンプレミスディクテーションv1WebSocketエンドポイント | はい | 番号 |
| /speech/recognition/interactive/cognitiveservices/v1 | CognitiveServicesのオンプレミスインタラクティブv1WebSocketエンドポイント |  |  |
| /speech/recognition/conversation/cognitiveservices/v1 | コグニティブサービスのオンプレミス会話v1WebSocketエンドポイント |  |  |

これは次の融合です。

* コンテナのディクテーションエンドポイントを試している人（どのようにしてそのURLを取得したのかわかりません）
* 1回目のパーティーは、コンテナ内の一つであるエンドポイント。
* 1回目の代わりのパーティエンドポイントの復帰speech.fragmentメッセージspeech.hypothesis3つのメッセージ番目の部分のエンドポイントは、ディクテーションエンドポイントのために戻ります。
* Carbonクイックスタートはすべての使用を開始しますRecognizeOnce（インタラクティブモード）
* Carbonは、speech.fragmentメッセージを必要とするメッセージについては、インタラクティブモードでは返されないと主張しています。
* リリースビルドでアサートが発生するカーボン（プロセスを強制終了）。

回避策は、コードで継続的な認識を使用するように切り替えるか、（より迅速に）コンテナー内の対話型または継続的なエンドポイントに接続することです。コードの場合、エンドポイントをhost:port/ Speech / Recognition / Interactive / cognitiveservices / v1に設定します

さまざまなモードについては、音声モードを参照してください-以下を参照してください。

### スピーチモード-インタラクティブ、会話、ディクテーション

**インタラクティブ**

* コマンドアンドコントロールシナリオを対象としています。
* Xのセグメンテーションタイムアウト値があります。
* 認識された1つの発話が終了すると、サービスはその要求IDからの音声の処理を停止し、ターンを終了します。接続は閉じられていません。
* 認識の上限は20秒です。
* 呼び出す典型的なCarbon呼び出しはRecognizeOnceAsyncです。

**会話**

* 長時間実行される認識を意味します。
* セグメンテーションタイムアウト値はYです。（Y！= X）
* ターンを終了せずに、複数の完全な発話を処理します。
* あまりにも多くの沈黙のためにターンを終了します。
* Carbonは、新しいリクエストIDを続行し、必要に応じてオーディオを再生します。
* 10分間の音声認識後、サービスは強制的に切断されます。
* Carbonは、未確認のオーディオを再接続して再生します。
* でCarbonで呼び出されStartContinuousRecognitionます。

**ディクテーション**

* ユーザーが句読点を話すことで指定できるようにします。
* 認識を開始するAPI呼び出しに関係なくEnableDictation、SpeechConfigオブジェクトを指定することにより、Carbonで呼び出されます。
* 1番目のパーティのクラスタ返すspeech.fragment中間結果についてのメッセージ、3番目のパーティのリターンspeech.hypothesisメッセージ。

適切な修正は、オンプレミスをサポートするSDK 1.8に付属しています（適切なエンドポイントを選択するため、オンラインサービスよりも悪くなることはありません）。それまでの間、継続的な認識のためのサンプルがありますが、それを指摘してみませんか？

<https://github.com/Azure-Samples/cognitive-services-speech-sdk/blob/6805d96bf69d9e95c9137fe129bc5d81e35f6309/samples/python/console/speech_sample.py#L196>

### さまざまなオーディオファイルにどのモードを使用する必要がありますか？

[Pythonを使用し](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/get-started-speech-to-text?pivots=programming-language-python)た[クイックスタート](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/get-started-speech-to-text?pivots=programming-language-python)は次の[とおりです](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/get-started-speech-to-text?pivots=programming-language-python)。ドキュメントサイトにリンクされている他の言語を見つけることができます。

インタラクティブ、会話、ディクテーションを明確にするためだけに。これは、サービスが音声要求を処理する特定の方法を指定する高度な方法です。残念ながら、オンプレミスコンテナの場合、完全なURIを指定する必要があるため（ローカルマシンが含まれているため）、この情報は抽象化から漏洩しました。今後、これをより使いやすくするために、SDKチームと協力しています。

### トランザクション/秒/コアの大まかな測定値をどのようにベンチマークできますか？

既存のモデルから予想される大まかな数値の一部を次に示します（GAで出荷するモデルではより良いものに変更されます）。

* ファイルの場合、スロットリングはスピーチSDKの2倍になります。オーディオの最初の5秒間は抑制されません。デコーダーは約3倍のリアルタイムを実行できます。このため、全体的なCPU使用率は、1回の認識で2コア近くになります。
* マイクの場合、リアルタイムの1倍になります。全体的な使用量は、1回の認識で約1コアである必要があります。

これはすべて、Dockerログから確認できます。実際には、セッションとフレーズ/発話の統計情報を含む行をダンプします。これには、RTF番号が含まれます。

### 同じホストで複数のコンテナを実行するにはどうすればよいですか？

ドキュメントは別のポートを公開するように言っていますが、LUISコンテナはまだポート5000でリッスンしていますか？

試してみてください-p <outside\_unique\_port>:5000。たとえば、-p 5001:5000。

## 技術的な質問

### 15秒未満の長さのオーディオを処理するために非バッチAPIを取得するにはどうすればよいですか？

RecognizeOnce()インタラクティブモードでは、発話が短いと予想される音声コマンドを対象としているため、最大15秒の音声のみを処理します。StartContinuousRecognition()口述や会話に使用する場合、15秒の制限はありません。

### 推奨されるリソース、CPUおよびRAMは何ですか。50の同時リクエストの場合？

4コア、4 GB RAMはいくつの同時リクエストを処理しますか？たとえば、50の同時リクエストを処理する必要がある場合、コアとRAMの数が推奨されますか？

リアルタイムでは、最新の8であるen-USため、6つの同時リクエストを超えるより多くのDockerコンテナーを使用することをお勧めします。16コアを超えるとクレイジーになり、不均一メモリアクセス（NUMA）ノードセンシティブになります。次の表に、各音声コンテナのリソースの最小割り当てと推奨割り当てを示します。

* [スピーチからテキストへ](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-faq?tabs=stt#tabpanel_1_stt)
* [カスタム音声テキスト変換](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-faq?tabs=stt#tabpanel_1_cstt)
* [テキスト読み上げ](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-faq?tabs=stt#tabpanel_1_tts)
* [カスタムテキスト読み上げ](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-faq?tabs=stt#tabpanel_1_ctts)

| **表3** | | |
| --- | --- | --- |
| **コンテナ** | **最小** | **推奨** |
| スピーチからテキストへ | 2コア、2GBメモリ | 4コア、4GBメモリ |

* 各コアは少なくとも2.6GHz以上である必要があります。
* ファイルの場合、調整はSpeech SDKの2倍になります（音声の最初の5秒間は調整されません）。
* デコーダーは約2〜3倍のリアルタイムを実行できます。このため、全体的なCPU使用率は、1回の認識で2コアに近くなります。そのため、コンテナインスタンスごとに3つ以上のアクティブな接続を維持することはお勧めしません。極端な側面は、のような8コアマシンに2倍のリアルタイムで約10個のデコーダーを配置することDS13\_V2です。コンテナバージョン1.3以降の場合、設定してみることができるパラメータがありますDECODER\_MAX\_COUNT=20。
* マイクの場合、リアルタイムの1倍になります。全体的な使用量は、1回の認識で約1コアである必要があります。

あなたが持っているオーディオの合計時間数を考慮してください。数が多い場合は、信頼性/可用性を向上させるために、ロードバランサーの背後で、単一のボックスまたは複数のボックスのいずれかで、コンテナーのインスタンスをより多く実行することをお勧めします。オーケストレーションは、Kubernetes（K8S）とHelmを使用するか、Dockercomposeを使用して実行できます。

例として、1000時間/ 24時間を処理するために、VMごとに10個のインスタンス/デコーダーを使用して3〜4個のVMをセットアップしようとしました。

### Speechコンテナは句読点をサポートしていますか？

オンプレミスコンテナで利用可能なキャピタライゼーション（ITN）があります。句読点は言語に依存し、中国語や日本語などの一部の言語ではサポートされていません。

私たちは、やる既存のコンテナの暗黙的および基本的な句読点をサポートしているが、それはあるoffデフォルトで。つまり.、例ではキャラクターを取得できますが、キャラクターは取得できません。。この暗黙のロジックを有効にするために、Speech SDKを使用してPythonでこれを行う方法の例を次に示します（他の言語でも同様です）。

Pythonコピー

speech\_config.set\_service\_property(

name='punctuation',

value='implicit',

channel=speechsdk.ServicePropertyChannel.UriQueryParameter

)

### 音声テキストコンテナにデータをPOSTしようとすると404エラーが発生するのはなぜですか？

HTTPPOSTの例を次に示します。

HTTPコピー

POST /speech/recognition/conversation/cognitiveservices/v1?language=en-US&format=detailed HTTP/1.1

Accept: application/json;text/xml

Content-Type: audio/wav; codecs=audio/pcm; samplerate=16000

Transfer-Encoding: chunked

User-Agent: PostmanRuntime/7.18.0

Cache-Control: no-cache

Postman-Token: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Host: 10.0.75.2:5000

Accept-Encoding: gzip, deflate

Content-Length: 360044

Connection: keep-alive

HTTP/1.1 404 Not Found

Date: Tue, 22 Oct 2019 15:42:56 GMT

Server: Kestrel

Content-Length: 0

どちらの音声テキストコンテナでもRESTAPIはサポートされておらず、SpeechSDKを介したWebSocketのみがサポートされています。常に公式ドキュメントを参照してください。[クエリ予測エンドポイントを](https://docs.microsoft.com/ja-jp/azure/cognitive-services/speech-service/speech-container-howto#query-the-containers-prediction-endpoint)参照してください。

### コンテナがroot以外のユーザーとして実行されているのはなぜですか？これが原因でどのような問題が発生する可能性がありますか？

コンテナ内のデフォルトユーザーはroot以外のユーザーであることに注意してください。これにより、コンテナをエスケープし、ホストノードでエスカレーションされたアクセス許可を取得するプロセスに対する保護が提供されます。デフォルトでは、OpenShift Container Platformなどの一部のプラットフォームは、任意に割り当てられたユーザーIDを使用してコンテナーを実行することにより、これをすでに実行しています。これらのプラットフォームの場合、root以外のユーザーは、書き込みが必要な外部にマップされたボリュームに書き込むためのアクセス許可を持っている必要があります。たとえば、ロギングフォルダやカスタムモデルのダウンロードフォルダなどです。

### 音声認識サービスを使用しているときに、なぜこのエラーが発生するのですか？

cmdコピー

Error in STT call for file 9136835610040002161\_413008000252496:

{

"reason": "ResultReason.Canceled",

"error\_details": "Due to service inactivity the client buffer size exceeded. Resetting the buffer. SessionId: xxxxx..."

}

これは通常、音声認識コンテナが取得できるよりも速く音声をフィードした場合に発生します。クライアントバッファがいっぱいになり、キャンセルがトリガーされます。音声を送信する同時実行性とRTFを制御する必要があります。

### C ++の例から、これらのテキスト読み上げコンテナのエラーについて説明していただけますか？

コンテナのバージョンが1.3より古い場合は、次のコードを使用する必要があります。

C ++コピー

const auto endpoint = "http://localhost:5000/speech/synthesize/cognitiveservices/v1";

auto config = SpeechConfig::FromEndpoint(endpoint);

auto synthesizer = SpeechSynthesizer::FromConfig(config);

auto result = synthesizer->SpeakTextAsync("{{{text1}}}").get();

古いコンテナには、CarbonがFromHostAPIと連携するために必要なエンドポイントがありません。コンテナがバージョン1.3で使用されている場合は、次のコードを使用する必要があります。

C ++コピー

const auto host = "http://localhost:5000";

auto config = SpeechConfig::FromHost(host);

config->SetSpeechSynthesisVoiceName(

"Microsoft Server Speech Text to Speech Voice (en-US, AriaRUS)");

auto synthesizer = SpeechSynthesizer::FromConfig(config);

auto result = synthesizer->SpeakTextAsync("{{{text1}}}").get();

以下は、FromEndpointAPIの使用例です。

C ++コピー

const auto endpoint = "http://localhost:5000/cognitiveservices/v1";

auto config = SpeechConfig::FromEndpoint(endpoint);

config->SetSpeechSynthesisVoiceName(

"Microsoft Server Speech Text to Speech Voice (en-US, AriaRUS)");

auto synthesizer = SpeechSynthesizer::FromConfig(config);

auto result = synthesizer->SpeakTextAsync("{{{text2}}}").get();

SetSpeechSynthesisVoiceName更新されたテキスト読み上げエンジンを備えたコンテナには音声名が必要なため、この関数が呼び出されます。

## 次のステップ