Azure TTS Integration - Implementation Guide

🎯 Overview

Your phone system now supports **Azure Neural HD voices** for significantly better voice quality compared to Vonage's built-in "Amy" voice.

Current Status

- ✅ Azure TTS service implemented
- ✅ Audio caching for performance
- ✅ Automatic fallback to Vonage TTS if Azure fails
- ⚠️ **Your current Azure key appears to be invalid/expired**

🔑 Update Azure Credentials

Step 1: Get Valid Azure Keys

1. Go to Azure Portal (https://portal.azure.com)
2. Navigate to **Cognitive Services** → **Speech Services**
3. Select your Speech resource (or create a new one)
4. Go to **"Keys and Endpoint"** section
5. Copy:
   - **Key 1** (or Key 2)
   - **Region** (e.g., `eastus`, `westus2`)

Step 2: Update Credentials

Run this command on the server:

```
cd /home/ubuntu/github_repos/auto_call_system
python3 update_azure_credentials.py
```

Or manually update the file:

```
nano /home/ubuntu/.config/abacusai_auth_secrets.json
```

Update the Azure section:

```
{
  "azure cognitive services": {
    "secrets": {
      "speech_key": {
        "value": "YOUR_NEW_AZURE_KEY_HERE"
      },
      "speech_region": {
        "value": "eastus"
      }
    }
  }
}
```

Step 3: Test the Service

Visit: `https://phone-system-backend.onrender.com/test/azure-tts`

You should see:

```
{
  "status": "success",
  "test_result": {
    "credentials_loaded": true,
    "region": "eastus",
    "service_status": "active",
    "test_audio_size": 24576
  },
  "available_voices": { ... }
}
```

🎙️ Available Voices

HD Neural Voices (Highest Quality)

1. **Andrew** (Male) - `en-US-AndrewMultilingualNeural`
   - Warm, professional tone
   - Best for: Business, customer service
   - Quality: HD (24kHz, 96kbps)

2. **Ava** (Female) - `en-US-AvaMultilingualNeural`
   - Friendly, engaging tone
   - Best for: Customer interactions, support
   - Quality: HD (24kHz, 96kbps)

Standard Neural Voices (Excellent Quality)

1. **Ryan** (Male) - `en-US-RyanMultilingualNeural`
   - Warm, approachable tone
   - Cost: Lower than HD
   - Quality: Neural (24kHz, 96kbps)

2. **Jenny** (Female) - `en-US-JennyNeural`
   - Clear, professional tone
   - Best for: Customer service

- Cost: Lower than HD
- Quality: Neural (24kHz, 96kbps)

🔄 How to Use Azure Voices

Method 1: Simple Speech (No Input)

```
from azure_tts_helper import create_azure_speech_ncco

ncco = create_azure_speech_ncco(
    text="Welcome to our sports facility!",
    voice='andrew',  # or 'ava', 'ryan', 'jenny'
    style='friendly'  # optional
)
return jsonify(ncco)
```

Method 2: Speech with Input (Question/Response)

```
from azure_tts_helper import create_azure_speech_input_ncco

ncco = create_azure_speech_input_ncco(
    text="What sport would you like to book?",
    context_state='awaiting_sport_selection',
    voice='ava',
    style='friendly'
)
return jsonify(ncco)
```

Method 3: Change Default Voice Globally

Edit `azure_tts_helper.py`:

```
# Change this line:
DEFAULT_VOICE = 'andrew'  # Change to 'ava', 'ryan', or 'jenny'
```

💰 Azure Pricing & Credits

Free Tier

- **0.5 million characters/month** for Neural TTS
- Approximately **~80,000 words** or **~40 hours** of speech
- Perfect for testing and small-scale deployments

Paid Tier

- **Standard**: $1 per 1 million characters
- **Neural**: $15 per 1 million characters
- **Neural HD**: $30 per 1 million characters

Check Your Balance

Visit: Azure Billing Portal (https://portal.azure.com/#view/Microsoft_Azure_Billing/BillingMenuBlade/~/Overview)

🔧 Current Implementation

What's Using Vonage TTS (Old)

Currently, **all 16 voice responses** in `app.py` use:

```
{
    "action": "talk",
    "text": "Some message",
    "voiceName": "Amy"  # ← Vonage built-in voice
}
```

What Will Use Azure TTS (New)

After migration, responses will use:

```
{
    "action": "stream",
    "streamUrl": ["https://phone-system-backend.onrender.com/audio/azure/xxxxx.mp3"]
}
```

📊 Features

✅ Implemented

1. **Azure TTS Service** (`azure_tts_service.py`)
   - Generates high-quality speech using Azure Neural voices
   - Automatic audio caching for performance
   - Token management (auto-refresh)

2. **Helper Functions** (`azure_tts_helper.py`)
   - Easy-to-use NCCO generators
   - Automatic fallback to Vonage if Azure fails
   - Voice switching support

3. **Audio Serving** (`app.py`)
   - `/audio/azure/<filename>` - Serves generated audio
   - `/test/azure-tts` - Tests Azure service

4. **Audio Caching**
   - Stores generated audio in `audio_cache/`
   - Prevents duplicate API calls
   - Saves costs and improves performance

🚀 Next Steps (Optional)

1. **Migrate All Voice Responses**
   - Replace all `create_speech_input_ncco()` calls with `create_azure_speech_input_ncco()`
   - Test each flow to ensure quality

2. **Voice A/B Testing**
   - Test Andrew vs Ava with real callers
   - Gather feedback on which sounds better

3. **Custom Styles**
   - Use `style='friendly'`, `style='cheerful'` for different contexts
   - Adjust `rate` and `pitch` for emphasis

🐛 Troubleshooting

Issue: “Authentication failed - Invalid API key”

Solution: Your Azure key is expired or invalid. Follow “Update Azure Credentials” above.

Issue: “Audio file not found”

Solution:
1. Check that `audio_cache/` directory exists
2. Verify permissions: `chmod 755 audio_cache/`
3. Check disk space: `df -h`

Issue: “Falling back to Vonage TTS”

Solution: This is normal! The system automatically falls back if Azure is unavailable. Check logs for the specific error.

Issue: “Service status: failed”

Solution:
1. Verify Azure credentials are correct
2. Check network connectivity to Azure
3. Ensure Azure Speech Service is enabled in your Azure portal

---

📝 Example Migration

Before (Vonage TTS):

```
def handle_greeting():
    return create_speech_input_ncco(
        "Welcome! How can I help you today?",
        'awaiting_intent'
    )
```

After (Azure TTS):

```
def handle_greeting():
    from azure_tts_helper import create_azure_speech_input_ncco
    return create_azure_speech_input_ncco(
        "Welcome! How can I help you today?",
        'awaiting_intent',
        voice='andrew',  # or 'ava'
        style='friendly'
    )
```

📞 Testing

Test 1: Check Service Status

```
curl https://phone-system-backend.onrender.com/test/azure-tts
```

Test 2: Make a Test Call

1. Call your Vonage number
2. Listen to the voice quality
3. Compare to previous calls

Test 3: Check Cache

```
ls -lh /home/ubuntu/github_repos/auto_call_system/audio_cache/
```

🎯 Summary

✅ **What's Done:**
- Azure TTS service fully implemented
- Helper functions for easy integration
- Audio caching and serving
- Automatic fallback to Vonage

⚠️ **What You Need:**
- Valid Azure Speech Service credentials
- Update the credentials using the guide above

🚀 **Next Steps:**
1. Update Azure credentials
2. Test the service (`/test/azure-tts`)
3. Choose your preferred voice (Andrew or Ava)
4. Optionally migrate all voice responses

📧 Need Help?

If you need assistance:
1. Check the test endpoint for detailed error messages
2. Review the logs in `app.py`
3. Verify your Azure subscription is active and has credits